

# **An Emergent Perspective on Interoperation in Systems of Systems**

David A. Fisher

*March 2006*

TECHNICAL REPORT  
CMU/SEI-2006-TR-003  
ESC-TR-2006-003





**Carnegie Mellon  
Software Engineering Institute**

---

Pittsburgh, PA 15213-3890

# **An Emergent Perspective on Interoperation in Systems of Systems**

CMU/SEI-2006-TR-003  
ESC-TR-2006-003

David A. Fisher

*March 2006*

**Integration of Software-Intensive Systems**

Unlimited distribution subject to the copyright.

This report was prepared for the

SEI Administrative Agent  
ESC/XPB  
5 Eglin Street  
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2006 Carnegie Mellon University.

#### NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (<http://www.sei.cmu.edu/publications/pubweb.html>).

---

# Table of Contents

<b>Acknowledgments .....</b>	<b>v</b>
<b>Executive Summary .....</b>	<b>vii</b>
<b>Abstract.....</b>	<b>xi</b>
<b>1 Introduction.....</b>	<b>1</b>
1.1 Context for this Report.....	1
1.2 Overview of this Report .....	2
<b>2 Systems of Systems .....</b>	<b>5</b>
2.1 Characterizing Systems of Systems .....	5
2.2 Implications for Systems of Systems .....	7
2.3 Inevitability of Systems of Systems .....	9
2.4 Scope of Systems of Systems .....	10
2.5 Interdependence of Systems of Systems .....	10
2.6 Natural Systems of Systems.....	11
<b>3 Emergent Behavior .....</b>	<b>13</b>
3.1 Influence .....	13
3.2 Cascade Effects and Epidemics .....	15
3.3 Emergent Composition .....	17
3.4 Emergent Properties .....	18
3.5 Coherent Structure .....	20
3.6 Tight Coupling.....	20
3.7 Semantic Issues .....	22
3.8 Implications for Emergence from Physics.....	23
3.9 Summary of Emergence .....	25
<b>4 Interoperation.....</b>	<b>27</b>
4.1 Integration vs. Interoperation .....	27
4.2 Scope of Interoperation .....	29
4.3 Node-Centric Perspective.....	30

4.4	Contextual Influences and Constraints .....	32
4.5	Unnecessary Coupling .....	35
4.6	Boundaries of Systems of Systems .....	37
4.7	Managing Emergent Behavior .....	37
4.8	Maximize Accuracy/Minimize Constraints .....	38
4.9	Modeling and Simulation .....	39
4.10	Trust .....	39
4.11	Summary of Interoperability .....	40
<b>5</b>	<b>Recommendations for Follow-On Work .....</b>	<b>43</b>
	<b>References.....</b>	<b>47</b>

---

## List of Figures

Figure 1: Derivation of Systems-of-Systems Characteristics .....	8
Figure 2: Applicability of Traditional System Engineering.....	9
Figure 3: Cascading Effects .....	16
Figure 4: The Vicious Cycle of Tight Coupling.....	21





---

# Acknowledgments

Thank you to all my Integration of Software-Intensive Systems (ISIS) initiative colleagues at the Carnegie Mellon<sup>®</sup> University Software Engineering Institute (SEI) for the many discussions, ideas, and notes that have helped shape this report. I am especially grateful to David Carney, Craig Meyers, Ed Morris, and Pat Place, who have debated many of the issues with me; to Lisa Brownsword and Jim Smith, who are using many of these ideas in the System-of-Systems Interoperability Practices (SoSIP); to Suzanne Garcia, who carefully reviewed the report and suggested many of its figures; and to Dennis Smith and Tricia Oberndorf, who provided the time and resources needed for the research.

I am also indebted to Ira Monarch for discussions that helped clarify several of the topics, to Howard Lipson who earlier contributed to the foundational work on emergence, and to Alan Christie and David Mundie who provided insight through their earlier experiments with emergent behavior using the Emergent Algorithm Simulation Environment and Language (EASEL).

---

<sup>®</sup> Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.



---

# Executive Summary

Systems of systems have unique characteristics that distinguish them from traditional monolithic systems. They offer potential benefits and new challenges not found in traditional systems. Among these benefits and challenges are

- new kinds and levels of complexity
- the pervasive presence of emergent behavior
- the ability to dynamically adapt to unexpected and unanticipated situations
- continuous execution over extremely long times and through many evolutionary cycles

Those characteristics of systems of systems derive from the operational and managerial independence of their constituent parts, from independent evolution, and from the character of emergent effects. In turn, those elements derive from the autonomy of the constituents, including (and especially) the human constituents. Systems of systems are the inevitable result of advances in computing and communications technologies and the growing expectations that accompany those advances.

Traditional monolithic systems depend on central control, global visibility, hierarchical structures, and coordinated activities as the primary compositional mechanisms to achieve their purposes. Those methods, however, rely on certain simplifying assumptions that do not apply in systems of systems. Consequently, many of the techniques and approaches of traditional software and systems engineering are ineffective and sometimes counterproductive in systems of systems. They are inadequate because they fail to address problems unique to autonomous constituents and emergent effects. They also are inefficient because they fail to exploit the advantages offered by adaptation and emergent behavior.

A system of systems depends on distributed control, cooperation, influence, cascade effects, orchestration, and other emergent behaviors as primary compositional mechanisms to achieve its purpose. New software and systems engineering methods are needed. Methods and approaches that manage emergent behavior and exploit emergent effects offer the possibility of cost-effective and predictable solutions in systems of systems.

Recognition of the importance of emergent effects in determining the global characteristics of systems imposes a change in perspective on the scope of a system. Traditional views that the software portions, computerized portions, or mechanized portions can be managed in isolation are no longer adequate. If a system is to fulfill its purpose, anything that significantly influences its resulting outcome must be viewed as part of the system. A system of systems does not stop at its software or mechanized portions but instead includes its acquirers, devel-

operators, users, sustainers, and others with direct impact on its behavior. Other influences include the business and legal environment, shared cultural characteristics, rewards and incentives, and levels of trust among the constituents.

Emergent behavior in the form of influence, indirect effects, cascades, and epidemics among the autonomous constituents permeates systems of systems. Emergent behavior is the inevitable consequence of the independent management, operation, and evolution that characterize systems of systems and is unavoidable in the presence of autonomous constituents. Influence and emergent effects are the only mechanisms by which autonomous constituents can cooperate to achieve their shared purpose, goals, or mission objectives. These effects produce emergent properties that cannot be localized to any single node or small number of nodes. Emergent properties in the form of products and services are the cumulative effects of the local actions and neighbor interactions of all the autonomous constituents.

Interoperation refers to cooperative interactions among loosely coupled autonomous constituents to adaptively fulfill system-wide purposes. These interactions enable emergent effects that produce the desired global properties in continuously changing situations. This contrasts with traditional integration processes that impose a composition through centralized control dependent on global visibility and coordination among predictable error-free components in predetermined situations. The effectiveness of interoperation depends on the degree to which the autonomous constituents share a common purpose and are able to individually act and interact in support of that purpose. Because emergent effects are involved, it is not necessary that actions be coordinated, that all constituents support all aspects of the purpose, or that any constituent function correctly all the time. There must be, however, sufficient cooperation and consistency of action to cause the desired system-wide products or services to emerge.

Effective methods are needed for generating and managing emergent effects with predictable results. Successful interoperation requires, among other things

- adopting a node-centric perspective that focuses on the system-wide implications of local actions
- avoiding assumptions that are invalid in systems of systems
- considering all influences that affect outcomes
- minimizing the number of constraints
- managing trust
- orchestrating successful outcomes

These principles must be extended to include not only more specialized techniques such as avoiding order  $n$ -squared computations and using adaptive optimization (as discussed in this report) but also approaches and techniques from biological and social systems, physical sciences, and other domains that demonstrate emergent behavior analogous to that of systems of systems.

Nevertheless, exploiting emergent behavior offers great potential, not only to overcome the problems of interoperation brought on by widespread use of systems of systems but also to achieve levels of adaptability, scalability, and cost-effectiveness not possible in traditional systems. Emergent methods offer possibilities for orchestrating solutions in which desired system-wide services are predictable consequences of cooperative local actions and interactions of individual autonomous constituents and for simplifying understanding of those solutions by focusing on, managing, and minimizing the number of constraints rather than concentrating on, managing, and minimizing the number of variables. Although there are no obviously insurmountable barriers to obtaining those benefits, much remains to be done to fulfill the promise of interoperability in systems of systems, with emergence at the center of both the problems and the solutions.



---

# Abstract

This technical report characterizes systems of systems from several perspectives; shows the role of emergent behavior in systems of systems; and introduces interoperability as the domain of development, use, sustainment, and evolution for systems of systems. It argues that the increasing importance of systems of systems was inevitable, emergent behavior is inherent in systems of systems, traditional software and systems engineering methods are inadequate for interoperation of systems of systems, and emergent methods offer a potential for cost-effective and predictable solutions. This report aims to facilitate discussion and reasoning about interoperation within systems of systems by showing some of the interdependencies among systems, emergence, and interoperation. It establishes a sizable but incomplete repertoire of topics, characteristics, and principles that are fundamental to the intersection of systems of systems, emergent behavior, and interoperation.





---

# 1 Introduction

At least informally, the concept of *systems of systems* is now widely recognized. In particular, there is broad recognition that many systems—including those that are network-structured, software-intensive, or geographically dispersed—are qualitatively different from traditional large-scale systems. Such systems are becoming exponentially more complex. They involve components that are independently managed and operated. They are critically dependent on other systems that are outside the administrative control of their owners, developers, and users. Their purpose, structure, and number of components are increasingly unbounded in their development, use, and evolution. Traditional systems engineering approaches and methods are often inadequate or inappropriate for systems of systems.

Greater understanding is needed regarding what distinguishes systems of systems from traditional monolithic systems, why those differences are arising now, and how they affect the acquisition, development, sustainment, and use of systems. Such understanding is needed as a foundation for developing approaches, processes, methods, tools, management techniques, policies, and technologies that will be effective in ensuring that systems of systems can be created, evolved, and used cost-effectively to fulfill real needs.

In contrast with traditional systems, systems of systems display *emergent behavior*. Emergent behaviors are actions that cannot be localized to any single component of the system but instead produce effects (often in the form of services) that arise from the cumulative action and interactions of many independently acting components. Emergence is the unavoidable result of interactions among autonomous entities and thus will occur in systems of systems whether by accident or intention. Emergence can be instrumental to both the success and failure of systems of systems.

*Interoperation* within systems of systems encompasses a variety of problems, solutions, relationships, and knowledge relevant to development, use, and evolution of systems of systems. These issues arise in the interactions between autonomous constituents of systems of systems and have few counterparts in the traditional integration of monolithic systems. Emergent behavior and interoperation offer different perspectives on the same issues.

## 1.1 Context for this Report

In this report, we attempt to provide a unified and consistent view of how systems of systems, emergent behavior, and interoperation relate to one another and to the practical aspects of creating and evolving real-world systems. This view encompasses a broad spectrum of existing knowledge, understanding, opinions, and intuitions about how systems of systems behave

in practice. It provides a foundation for reasoning and research in interoperation and emergence. It introduces a broad sample of topics and issues relevant to systems of systems, interoperation, and emergent behavior.

The ideas reported here derive from ongoing work by the Integration of Software-Intensive Systems (ISIS) initiative at the Carnegie Mellon<sup>®</sup> University Software Engineering Institute (SEI), earlier research at the CERT<sup>®</sup> Coordination Center (CERT/CC) also at the SEI, and an extensive open literature addressing complex systems under a variety of names. ISIS has been examining several aspects of interoperation in systems of systems, including:

- examination of perspectives on interoperation and systems of systems [Brownsword 04]
- investigation of the dimensions that may be relevant to interoperation within systems of systems [Morris 04]
- identification of characteristics and approaches to interoperability [Carney 05a]
- analysis of processes and tools that may be useful in addressing problems within systems of systems [Lewis 04b]
- interoperability in acquisition [Meyers 05]
- role of semantics in systems of systems
- issues related to evolution in systems of systems [Carney 05b]

Previous CERT/CC work was aimed primarily at survivability and infrastructure assurance in networked and unbounded systems with special emphasis on critical national infrastructures such as the Internet and the electric power grid. That research laid the groundwork for understanding, reasoning, and experimenting with emergent phenomena. We developed automated tools for accurate but imprecise simulation of systems of systems [Christie 03] and made extensive use of discrete-event, also called agent-based, simulation to better understand emergent behavior. More recent work with a major defense program provided practical insight into an evolving large-scale operational system of systems in a specialized application domain.

## 1.2 Overview of this Report

The concepts of systems of systems, emergence, and interoperation are bound up in one another. Emergence can exist only within a system of systems and is the dominant mechanism for determining the outcomes of such systems. Interoperation, also called interoperability, has to do with the exchange and use of information necessary for effective operation of a system of systems. It includes problems, solutions, and relationships important to systems of systems. Interoperation encompasses the understanding, know-how, techniques, methods, meas-

---

<sup>®</sup> Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University. CERT and CERT Coordination Center are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University

ures, and tools that allow orchestration and exploitation of emergent effects to fulfill the global objectives of systems of systems.

This technical report serves as a brief introduction to concepts that characterize systems of systems, emergence, and interoperation. It describes the relationships among those concepts and gives an indication of their implications. It does not provide specific techniques or methods for addressing interoperation in systems of systems. It is our hope that this report will stimulate interest in the development of sound theory and drive the development of effective practices for interoperation.

Sections 2, 3, and 4 explain the general concepts of systems of systems, emergence, and interoperation, respectively, and the relationships among them. Section 4 also points out some promising approaches to interoperation in systems of systems. Section 5 identifies a broad spectrum of topics and issues that are relevant to interoperation and emergence but beyond the scope of this report.



---

## 2 Systems of Systems

Systems of systems have been recognized as a distinct class of system for nearly a decade. The intuitive idea has been that certain modern systems display kinds and levels of complexity not previously encountered in automated and software-intensive systems and that this complexity results in unanticipated negative behavior ranging from surprising mismatches through catastrophic local failures to completed systems that cannot fulfill real needs. Furthermore, rigorous and intense application of traditional management and systems engineering methods not only is ineffective but often aggravates the problems.

### 2.1 Characterizing Systems of Systems

Attempts have been made to characterize systems of systems by enumerating some of their more salient properties. These might include some combination of the following terms: large, networked, unbounded, geographically distributed, having complex internal interfaces, adaptive, dynamic, evolving, without global visibility, interdependent, distributively controlled, emergent, and nonhierarchical. Although each of these characteristics can be found in systems of systems, most of them can also be found in some monolithic systems. Furthermore, not all of them are present in every system of systems. Maier and others combine five properties to characterize systems of systems as those that have

1. operational independence
2. managerial independence
3. evolutionary development
4. emergent behavior
5. geographic distribution [Maier 98]

Although some systems of systems do not have all of them, most of these properties are unique to systems of systems, especially if one is careful when drawing the boundaries of a system.

To have operational and managerial independence, one of two approaches must prevail: the operational personnel and managers must be considered as part of the system, or operations and management must be automated. Traditionally, systems were often considered to encompass only the automated and mechanized components. In practice, people were left out of the equation. By independence of operations and management, we mean that the individual constituents of the system are able to act independently. It is this independence that distinguishes systems of systems. Traditional monolithic systems depend on centralized control, global

visibility, and hierarchical structures—none of which is fully achievable in the presence of independent management and operations. The presence of independent management and independent operation—combined with reduced visibility and reduced effectiveness of centralized control and of hierarchical structures—serves to increase complexity and reduce the appropriateness of traditional tools that depend on assumptions of centralized control, global visibility, and hierarchical structure.

Evolutionary development in systems of systems is independent, explicitly recognized, and continuous. All useful systems evolve, but in traditional monolithic systems, evolution has seldom been treated as an integral aspect of the design, implementation, management, and operational process. In systems of systems, the management and operational independence of the constituents enables their independent evolution. This independence of change in individual constituents adds significantly to the complexity of the interactions among constituents and of management and operations. Thus, in systems of systems, evolution must be explicitly recognized and managed. Explicit recognition encourages use and exploitation of evolution and, therefore, more frequent changes. Even without increased frequency of change in individual constituents, evolution will appear more continuous from a global perspective, due to the lack of system-wide coordination of evolutionary changes.

As separated constituents manage their local domains in ways most advantageous to themselves and to fulfilling their commitments to the system as a whole, geographic distribution and networking of systems encourages independent management, operations, and evolution. Geographic distribution reduces visibility and thus the effectiveness of centralized control. It also encourages a nonhierarchical networked structure whose topology is strongly influenced by the relative geographical positions of the constituents. Although geographic distribution tends to enable local autonomy and engender systems of systems, some geographically distributed systems can approximate the assumptions necessary for monolithic systems. Independence of management, operations, and evolution—as well as all of the complexities of systems of systems—can occur without geographic distribution. Thus, geographic distribution is neither necessary nor sufficient to characterize systems of systems.

Emergent behavior, in one sense, best distinguishes systems of systems because it is the one characteristic always present in systems of systems and never present in monolithic systems. Although emergence is important in developing, managing, and evolving systems of systems, emergence does not provide a good test for identifying systems of systems because it is difficult to determine whether a system-wide property was generated by emergent behavior.

Although Maier's five characteristics provide a reasonable intuitive notion of systems of systems, we need, instead, a characterization that distinguishes between monolithic systems (for which traditional systems engineering and management approaches are appropriate) and systems of systems (that display the kinds and levels of complexity for which traditional methods are inadequate and were never intended). We also need a characterization from which the observed characteristics can be derived and explained.

Our approach is to ask what gives rise to the management independence, operational independence, evolutionary independence, and emergent behavior that generate the kinds and levels of complexity observed in systems of systems. All of these characteristics derive from the presence of autonomous constituents in the system. Individual constituents may be automated, mechanized, or human. Without their presence and autonomy, the independence and emergent behavior cannot arise. Furthermore, monolithic systems cannot have autonomous constituents, or they would not be monolithic. At the same time, the hierarchical structures, centralized control, tight coupling, and closed-system constraints of monolithic systems are intended to prevent autonomous actions by individual components. The presence of autonomous constituents is both necessary and sufficient to characterize systems of systems.

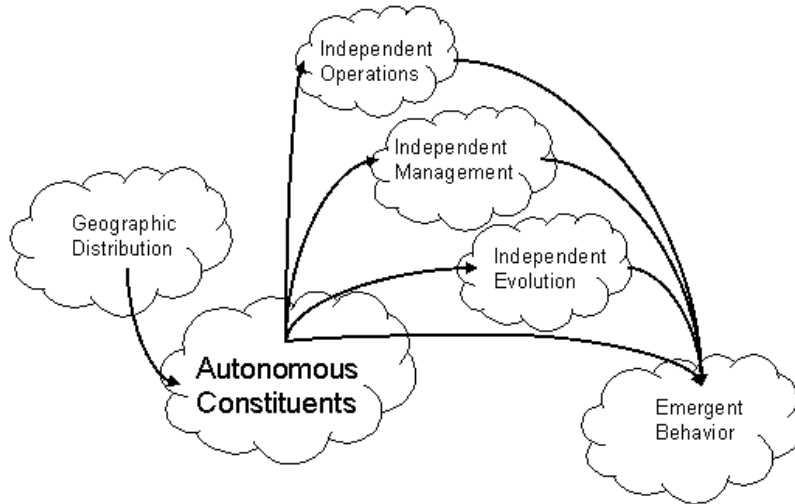
A *system of systems* is any system composed of systems that are themselves autonomous. By *system* we mean any interacting or interdependent group of entities that forms a unified and purposeful whole. By *autonomous* we mean that an entity can exercise independent action or decision making. For example, an automobile is generally viewed as nonautonomous because it is thought to be under the control of its driver. An unmanned vehicle is autonomous if it can take independent actions that are influenced by the dynamic conditions of its environment without human intervention but not if its actions are remotely controlled. In general, a system is autonomous if, and only if, it can take actions that are influenced by factors other than its design and externally specified parameters. These factors might include its independent decisions, external influences not included in its parameters, and the influence of component failures, accidents, design flaws, or user errors. Hereafter, the term *constituent* will be used only when referring to an autonomous component of a system of systems.

## 2.2 Implications for Systems of Systems

From the preceding characterization of systems of systems, it follows that any system with operational independence, management independence, or emergent behavior will be a system of systems because each of these characteristics involves the presence and participation of autonomous components. Autonomous components provide strong incentive for independent action in management and operations, while emergent effects arise from combinations of independent actions. Thus, operational independence, managerial independence, and emergent behavior are both uniquely and universally characteristic of systems of systems.

Because almost all systems evolve in response to changing needs and technological advances, the fact of evolutionary development alone cannot distinguish a system of systems. Systems of systems, however, are unique in that their autonomous components can evolve independently of one another. Without knowledge of how their neighbors are evolving, constituents are likely to evidence incompatibilities, with unanticipated and unintended effects. This creates a level of complexity in the evolution of systems not found in monolithic systems. Although systems of systems need not be geographically distributed, this characteristic encourages local autonomy, which can spur the independence in operations, management and evolution that typifies systems of systems.

This perspective on systems of systems is summarized in Figure 1, where it can be seen that four of the five Maier characteristics derive from the autonomy of the constituents, geographic distribution encourages autonomy, and emergence derives directly from autonomy as well as from the other Maier characteristics.



**Figure 1: Derivation of Systems-of-Systems Characteristics**

With respect to monolithic systems, the view provided thus far is somewhat idealized. The characteristics described for systems of systems have been long observed but usually can be safely ignored in monolithic systems. That is, independent operations, management, and evolution, and, in fact, emergent effects have at times been observed in what have been traditionally called monolithic systems. Such effects generally have been sufficiently insignificant that they can be ignored. It truth, most real systems satisfy the necessary and sufficient properties for a system of systems. Thus, as a practical matter, a *monolithic system* is any system for which systems-of-systems characteristics are either absent or have sufficiently insignificant influence on outcomes that they can be ignored. In particular, in systems where it is safe to assume the presence of characteristics such as global visibility, effectiveness of central control, and hierarchical structures and the absence of emergent effects and unknown external influences, it is appropriate to use traditional software engineering methods, approaches, and tools that depend on those assumptions. It follows then, from a pragmatic perspective, that it is *unsafe* to embrace the assumptions of monolithic systems for any system in which emergent effects are sufficiently important in influencing outcomes not to be ignored. The latter two points are illustrated in Figure 2. It is the presence of autonomous constituents that makes emergent behavior and systems of systems possible. It is the dominance of autonomous constituents that generates emergent behavior and requires that systems be treated as systems of systems.



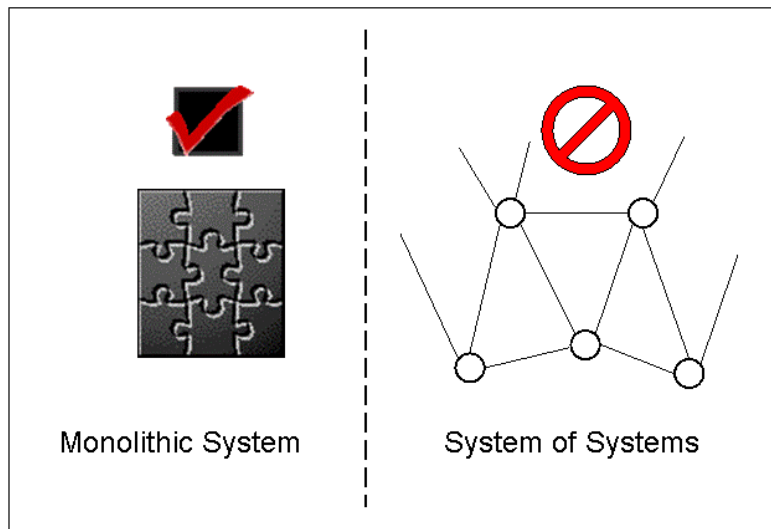


Figure 2: *Applicability of Traditional System Engineering*

## 2.3 Inevitability of Systems of Systems

Only in the last few decades have automated systems of systems been recognized as a distinct class of systems. Systems of systems are the inevitable consequence of advances in communications and computing technology. Improvements in communications bandwidth, reliability, and cost-effectiveness have allowed systems to be interconnected and to become interdependent in ways not possible in standalone systems. The advent of networked systems without hierarchical structure allows larger numbers of components, more numerous and complex interconnections, and greater geographical distribution than were previously possible. Often when a monolithic system joins a network, it retains its autonomy with respect to the rest of the network. Thereby, individual monolithic systems become autonomous constituents of a system of systems that is the network.

Advances in computing technology have allowed the control sections of mechanical, electrical, and electrical-mechanical machines to be replaced by software running on general-purpose computing devices, turning those machines into software-intensive systems—and in some cases into autonomous systems. An *autonomous system* is a system that takes independent action or makes independent decisions with respect to the system of which it is a part. In its internal structure, an autonomous system can be either monolithic or a system of systems. When systems are implemented on general-purpose computing devices, only discipline in their development and management prevents them from becoming autonomous. The obvious benefits of combining existing, often autonomous, systems and of giving greater autonomy to component devices conspire to continually increase the size, numbers, and complexity of systems of systems.

The autonomy of components itself also offers significant advantages. Each constituent can be designed, implemented, tested, and evolved independently of the systems in which it will

be used. This independence reduces the amount of information that constituents must have about each other and simplifies the system as a whole. Just as importantly, it reduces or eliminates the cost and complexity of coordination among components. In addition, it allows components to be developed in parallel and to evolve without synchronization. Similarly, the independence of autonomous components increases the likelihood that they can be used in multiple systems of systems.

Finally, the growing desire for scalable and adaptable systems necessitates an increased use of systems of systems. Adaptable systems are able to adjust roles and functionality of their components, quality of service, network structure, or other architectural characteristics to fulfill continuously changing needs. To be scalable, a system must be able to dynamically incorporate arbitrary numbers of additional components. Monolithic systems seldom can be either adaptable or scalable. Conversely, the autonomy of constituents enables and encourages the development of adaptable and scalable systems. Only through adaptability and scalability can systems simultaneously remain continuously executing and evolve to satisfy changing needs or to exploit technological advances.

As the expectations for and potential benefits of systems of systems grow, so does the demand for such systems. Their number will continue to increase and their importance to intensify. Nowhere is this acceleration more obvious than in the U.S. Department of Defense, where there is a rising advocacy for transformation, driven by technological advances in computing and communication and instantiated in a vision of system of systems known as network-centric warfare (NCW) [Alberts 99].

## 2.4 Scope of Systems of Systems

The owners, developers, users, and other stakeholders of traditional monolithic systems have typically been viewed as separate and apart from the system. However, the adaptive, emergent, and evolving character of systems of systems means that their behavior changes continuously in response to the influence of stakeholders. Even the claim that systems of systems display management and operational independence conveys the perspective that managers and operational users are integral to the system. Those that create, manage, use, own, evolve, or influence the outcomes of a system of systems must be viewed as constituents within the scope of concern for that system; otherwise, the outcomes will be determined by influences beyond the scope of concern and will not be predictable from an understanding of the system. Hereafter in this report, human constituents of a system of systems will be called *stakeholders*. Autonomous components or constituents will sometimes be called *nodes*, particularly when the system is viewed as a network.

## 2.5 Interdependence of Systems of Systems

Unlike traditional monolithic systems, systems of systems do not in general depend on assumptions of infinitely reliable components, complete global visibility, or absence of design,

implementation, and user errors. Even in the presence of unanticipated events, systems of systems are expected to survive and to contribute to their global objectives. The actions and interactions of components of systems of systems can be influenced by events external to the system, including aspects of their environment. Thus, systems of systems are always dependent on the influences of other systems of which they are a part. In this sense, they are also unbounded [Fisher 99]. Because the influence is in both directions, they are always interdependent with external systems.

## 2.6 Natural Systems of Systems

Like automated systems of systems, natural systems of systems—social, economic, and biological—are composed of autonomous constituents. They display the operational independence, evolutionary nature, and emergent behavior that characterize automated systems of systems. Natural systems also conspicuously lack the central control, global visibility, synchronous operation, coordinated interactions, and hierarchical structures that dominate traditional monolithic systems and systems engineering methods. Natural systems offer a repertoire of methods and approaches that may be adaptable to, or have analogies in, automated systems of systems.

To the extent that systems include human constituents, they are social systems. Thus, if an automated system is taken to include its owners, developers, or users, it is also a social system with all the problems and benefits that designation entails. The field of software engineering is built on a recognition of the importance of human activities in the acquisition, development, operation, and evolution of software-intensive systems.

Natural systems also provide insight into the nature of complexity in systems of systems. Like automated systems, natural systems (especially biological systems and systems of social insects) are often extremely complex when viewed in terms of their number of constituents, the dynamic system-wide structure of their interconnections, the enormous number of possible combinations of interactions, and the consequences of unanticipated external influences. They are, however, relatively simple when viewed in terms of the rules of behavior that determine the local actions and neighbor interactions of individual constituents and the global properties that will predictably emerge from the cumulative effects of those actions and interactions. The perceived complexity of a system as a whole arises from attempts to understand the enormous numbers of possible paths by which the global properties might arise. Perhaps the perceived complexity of automated systems of systems can be overcome by focusing on the local actions and interactions of constituents and understanding more clearly the emergent processes that will predictably produce desired global properties to satisfy system-wide goals.



---

## 3 Emergent Behavior

Emergent behavior is often observed but poorly understood, especially in the context of automated systems. Conceptually, emergent behavior refers to actions of a system as a whole that are not simple combinations of the actions of the individual constituents of the system. More precisely, systems of systems display certain global properties that cannot be accounted for as the result of preserving and combining actions and properties of their constituents. Emergent properties can take the form of quality attributes such as reliability, performance, safety, color, or texture. Alternatively, they can take the form of system-wide services such as message delivery in a communications network (see Section 3.4) or adequate power generation in an electric power grid. Thus, for example, when a highway becomes congested during rush hour and all traffic moves slowly, the slow movement of traffic is a global property of the highway system. The slow movement cannot be explained as a particular combination of actions of individual vehicles; instead, it arises from the cumulative effects of the actions and interactions of all the vehicles. It does not depend on the specific actions of the individual vehicles, and no individual vehicle plays a critical role. Furthermore, if some subset of the vehicles acted differently in their local actions (within certain boundaries), the global effect of slow-moving traffic would be unchanged. The resulting global effects cannot be accounted for by the individual actions of particular vehicles; instead, they depend on the general activities of sufficiently many of them within the context of that highway.

Because we don't understand enough about the processes by which local actions and interactions with neighbors are composed to produce emergent behavior, we often are surprised at the resulting emergent global effects. This has encouraged the belief that emergent behavior is synonymous with unexpected, unanticipated, unpredictable, and undesirable behavior. However, from the rush-hour example, the emergence of slowness of the traffic is highly predictable as a function of the number of vehicles involved. As will be seen below, emergent behavior arises naturally and predictably from influence mechanisms, cascade effects, and other emergent phenomena that are inherent in systems of systems.

*Emergence* or *emergent behavior* refers to indirect influences, cascade effects, and other processes that produce emergent properties. (For more on emergent properties, see Section 3.4.) Emergence also refers to emergent global properties that take the form of system-wide products or services.

### 3.1 Influence

Autonomous entities are capable of independent action, independent decision making, and self-direction. Where an entity is autonomous, it can only be influenced, not controlled, by

outside forces. *Influence* is any mechanism by which one entity interacts with another in a way that changes the physical, informational, or emotional state of the other. Influence can be negative as well as positive. Whether an influence is positive or negative is not inherent but instead depends on the perspective of the observer. Influence can be cooperative, adversarial, or neutral. Influence can be used with friends, enemies, or third parties to gain support for one's own cause.

A person, by definition, is autonomous. Criminal law, seen as an example of an outside force, does not control a person's behavior. For most individuals, the influence of the law in conjunction with other societal influences is usually sufficient to ensure their abiding by it. Because autonomous entities are capable of independent action and decision making, they will at times exercise that independence, especially when there are more local or more immediate conflicting rewards. At no time can one guarantee the independent action of an autonomous entity. In this sense, autonomous entities cannot be controlled. They can only be influenced in their decisions and actions.

Because people are autonomous entities, they can engage in agreements such as contractual requirements, laws, regulations, standards, mutual assent, unity of opinion, or harmony of intent. Agreements can be formal or informal. Agreements, however, are never absolute because people are subject to opposing influences. For example, a U.S. Government contractor on a cost-plus contract may have incentive to encourage changes that add new features and delays that will lead to cost-overruns, while one on a fixed-price contract may have incentive to encourage reductions in scope or functionality. On a larger scale, when an electric utility promises to provide continuous electric power to a city, it intends to do so only to the extent that more powerful influences do not intercede—influences such as damage by a natural disaster, blackouts induced by grid failures, total demand that exceeds planned capacity, or equipment failures resulting from cost-saving decisions to forgo preventive maintenance.

Agreements are always negotiated in the context of influences. In some cases, each side presents its wants and offers, and the two sides negotiate an agreement—giving each other inducements to consent. Other agreements, especially those in the form of laws and regulations, are determined by a legislative body far removed from the individuals to whom they apply. Though removed, legislators and regulators are strongly influenced by a combination of public, expert, and special-interest opinions. Furthermore, if there is strong public sentiment against it, a law or regulation will be ignored to the point of ineffectiveness, or public pressure will be applied to force changes. When drivers exceed the speed limit in a 55 mile-per-hour zone they are, in fact, negotiating with police for an enforced speed limit that is higher than the one prescribed by the regulators. If they are seldom ticketed, they have prevailed in the negotiation.

More to the point, any agreement is effective only to the extent that the parties intend to keep it and are capable of abiding by it. Each party has a set of intentions that reflects its own goals and objectives with respect to the agreement. Each also has expectations that reflect its perception of the other's intentions. A combination of extraneous influences, lack of capabil-

ity, and misrepresentation can ensure a mismatch between intentions and expectations, either or both of which may fall short of the stated agreement. Failure to fulfill agreements undermines both success and trust. Lack of trust often leads to reduced expectations, overstatement of needs, and exaggeration of capabilities in subsequent negotiations. The net effects are higher costs, extended schedules, and lower performance of systems—coupled with even greater loss of trust and cooperation. What actually happens in the development, operation, or evolution of a system is determined by the influence of all these considerations. Agreements are never controlling.

Influence, then, is the underlying mechanism for all interactions among autonomous entities. Because they cannot control one another, autonomous entities can achieve goals that are not local to themselves only by increasing their influence through cooperative interactions with others. For an autonomous entity, cooperation can arise through its own independent choice or direct influence by neighbors. Even independent choice, however, is influenced by the current state of the entity, which is itself the cumulative result of past influences. Thus, all actions and interactions by an entity are ultimately affected by its history of direct and indirect influences.

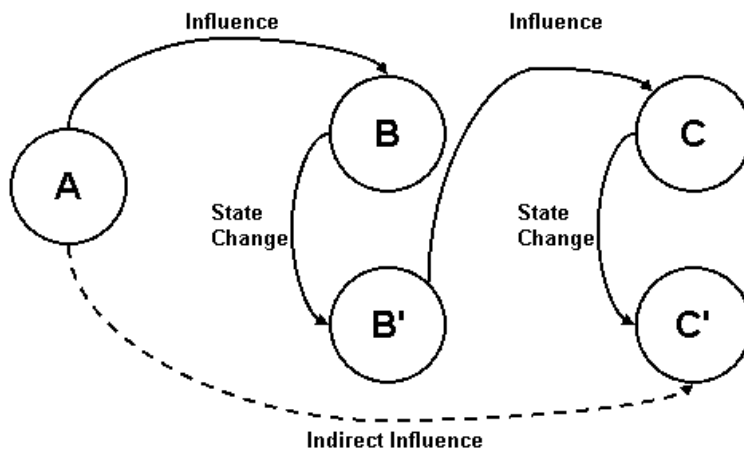
When constituents have opposing goals, they may negatively influence each other, knowingly or unintentionally, to further their own goals. For example, a stock racing car is designed, among other purposes, to perturb the air behind the car in ways that will destabilize cars following it during a race.

Given the significance of influence, centralized control can have only limited effectiveness in a system of systems where each component system is an autonomous entity. While influence restricts the imposition of external (including centralized) control, the lack of global visibility in systems of systems impairs attempts to validate compliance. In monolithic systems, synchronization and coordination among parts have been the primary means of imposing centralized control. However, coordination among parts makes systems brittle, unable to adapt to changing circumstances or unanticipated influences, and subject to accidents or failures in response to external influences. Centralized control is both ineffective and undesirable with regard to emergent effects in systems of systems. The alternative, orchestration, is discussed in Section 4.7.

## **3.2 Cascade Effects and Epidemics**

Emergent behavior arises from influence relationships through two primary mechanisms: cascade effects and emergent composition. Emergent composition, which will be discussed in Section 3.3, is the means by which influences in the form of local interactions are combined to generate properties or characteristics that cannot be derived from simple summations or combinations of the properties of their constituents. Cascade effects are the means by which influence and emergent effects are propagated throughout a system of systems.

*Cascade effects* are any succession of state changes in a sequence of entities generated from a single initial influence (Figure 3). Often the influence exercised by an entity, A, when interacting with another entity, B, will take the form of state changes in B that influence B's interactions with a third entity, C. In this way, A's actions indirectly influence C, after some time delay. Sequences of indirect influences are potentially arbitrarily long. A cascade effect occurs whenever such indirect influences form a chain involving two or more influence links. Furthermore, the kinds of properties that are affected and the magnitudes of those effects can vary at each step in the chain. A budget cut in one node might reduce quality of service to another, which might cause a schedule delay in a third node, which might impose significant costs on a fourth node.



**Figure 3: Cascading Effects**

Cascade effects are both inherent and pervasive in the interactions among constituents of a system of systems. They are inherent because interactions are essential to a set of entities constituting a system. Cascade effects are pervasive because any interactions cause state change and some portion of those state changes will affect future interactions.

Cascade effects can be amplified or dampened at each step of the chain with respect either to the number of entities that are influenced or to the degree of influence on individual constituents. In most cases, there is a natural tendency toward dampening at each step as existing states dominate over new influences. By this means, the number of nodes involved at subsequent steps can quickly be reduced to zero. Cascade effects of this kind have minimal global effect.

An epidemic is a special form of cascade effects that breaks their natural tendency toward dampening. An *epidemic* occurs when the number of constituents that are influenced increases at each step. Indeed, epidemics of diseases occur whenever the number of infected persons increases exponentially as a function of time. No epidemic, though, can continue indefinitely to grow in size or intensity. Every epidemic will end eventually, because of organized resistance, resource limitations, or saturation of its potential audience. It follows, then,



that epidemics can be stopped by removing potential nodes from the chain or by preventing nodes in the path from changing state. Note that it is not necessary to isolate or immunize every node, only a sufficient number in the path to adequately dampen the cascade effect. There can be epidemics of ideas as well as physical epidemics. The change to pervasive use of cell phones by young people in ways poorly understood by their parents is an epidemic of ideas. So too are fads.

Like those of influence, the results of cascade effects, epidemics, and cascading cycles

- can be viewed as positive, negative, or neutral
- can often vary among constituents
- may differ between local and global objectives

Cascades are inherent and critical to both the successes and failures of systems of systems.

Epidemics often have a tipping point [Gladwell 00]. It occurs when the epidemic effects seem to appear suddenly or unexpectedly in a large number of constituents. Consider a city with large amounts of trash strewn about. If every day a person picks up one piece of trash and convinces one other person to do likewise beginning the next day, then the number of pieces picked up will be one, two, and four respectively on the first, second, and third days. Such small amounts will certainly go unnoticed. However, if this process continues, eventually there will be a day in which, say, an eighth of the total trash is removed. Even then, the cleanup may escape notice of most of the public. Three days later, however, all of the trash will appear suddenly to have been removed, to the great surprise of many. The surprise might have been even greater if the interval was a month instead of a day, but the same purpose would have been achieved in about the same number of steps with about 30 times the delay. Tipping points are likely in any epidemic with exponential growth regardless of the delay between steps.

Chains of cascade effects can also form cycles, as would be the case if A and C were the same node in Figure 3. As in Systems Dynamics [Forester 61] and Systems Thinking [Senge 94], cyclic cascade effects can produce either reinforcing (amplifying) or balancing (stabilizing) loops. A reinforcing loop incrementally moves the state of the involved nodes (and indirectly often the system as a whole) in a particular direction. Reinforcing loops create a spiral of effects that are often interpreted as success or failure. Reinforcing loops, however, cannot continue indefinitely because they also produce secondary effects in the form of balancing loops. Each reinforcing loop ultimately must be consumed by a balancing loop.

### 3.3 Emergent Composition

An *emergent composition* is a mechanism by which the effects of autonomous entities are combined to produce configurations or patterns that cannot be expressed as a simple summation or combination of their parts. The configurations or patterns may be physical, biological, psychological, or symbolic. The configurations and patterns often take the form of services

and are normally characterized by the properties of those services. Because such properties arise through emergent composition, they are called emergent properties.

Emergent composition is often poorly understood and sometimes misunderstood because it has few analogies in traditional systems engineering. The erroneous view that emergence is unpredictable (and thus undesirable) arises, at least in part, from the difficulty in understanding how a quality attribute can arise through composition from parts that do not possess that attribute. Emergence of this kind is exemplified by building reliable systems from unreliable components.

An example of a biological pattern would be the efficiency of ants foraging for food. The individual ants are very inefficient; they take nearly random walks when no food has been found and exhibit frequent deviations from the most efficient of known paths even when the location of food is known. Yet, with these obvious inefficiencies plus the appropriate use of pheromones (which are used to mark trails), most individual ants will take a nearly optimal round trip path between the food source and their nest. Although the initial path is almost never optimal for a given situation, over time the path of each ant tends toward optimal. The larger a food source, the longer the time before it is exhausted; and thus the more nearly optimal the average path that will be taken to it.

There is always a tradeoff between adaptability and efficiency. In natural systems and in many automated dynamic systems, optimization can provide only a short-term advantage, while adaptability is a long-term necessity for survival. Any fixed optimization in traditional systems engineering can provide great efficiency for the exact circumstances for which it is intended. At the same time, the optimization undermines adaptability and is inefficient for other situations. In a dynamic system, other situations will arise, and what was optimal can become very inefficient. The most efficient system of systems is not the one that is optimized for its most commonly expected situation; it is the one that continuously adapts to improve its efficiency with respect to its current situation. Not only will it likely be more efficient on the average than any fixed optimization, but also its efficiency does not depend on precise or even accurate prior determination of what situations will arise.

As a general rule, it is best to opt for adaptability over optimization—but only in contexts where either the environment or the needs will likely change. With software, it is possible to design for adaptability, then to optimize dynamically for a situation that is actually encountered, and later to back out of the optimization when the situation changes.

### 3.4 Emergent Properties

Emergent composition is probably the most interesting and important mechanism for creating emergent properties. An *emergent property* is any characteristic of a system that cannot be localized to a single independently acting constituent or to a small constant number of constituents. Emergent properties arise from the cumulative effects of the local actions and neighbor interactions of many autonomous entities. The simplest kind of emergent property

involves concepts that are not meaningful in the context of a single constituent or small constant number of constituents. Emergent properties arising from cascade effects are of this kind. Epidemics and fads extend cascade effects to involve enough constituents to produce emergent properties of a kind that are characterized by their pervasiveness.

Another kind of emergent property takes the form of global services that are meaningful only when they involve a significant portion of the network nodes or system constituents. Internet Protocol (IP) routing in the Internet is a particularly good example of an emergent global service. No IP router knows the complete topology of interconnections for the Internet or even the configuration of local interconnections in its own neighborhood. Because the configuration of links among routers changes continuously, as does the available bandwidth on a given link, routing tables always correspond to an earlier configuration. And yet, IP routing is a reliable and efficient process that predictably gets messages from their source to their intended destination. Each IP router along the path of a message decides which of its immediate neighbor routers will constitute the next hop without knowledge of routers or likely paths beyond that immediate neighbor. IP routing, like most emergent services, must operate with incomplete, imprecise, and outdated information; nevertheless, it is able to provide efficient and predictable functionality.

IP routing implementations do not guarantee optimal paths, but they do predictably generate correct paths that satisfy affordability constraints (including those detailed in Section 3.8). In particular, they generate paths whose lengths are strictly less than order  $n$  where  $n$  is the total number of possible destinations. The Internet is subject to and must be able to dynamically adapt to accidents, user errors, equipment failures, natural disasters, and attacks by intelligent adversaries. IP routing manages this tradeoff between performance and adaptability in a way that, while adaptable and suboptimal, is always scalable and affordable without risk of local routing errors cascading into system-wide failures. This contrasts with the electric power grid where issues of local and global performance are often in conflict, leading at times to widespread power outages.

A more complex kind of nonlinear emergent property arises from conflicts between competing local objectives. The nonlinear effect occurs whenever changes in the value of a variable that characterizes an emergent property vary more than linearly with respect to a controlling variable of the emergence. An example is the pressure for widespread use of recording tapes that built up for years but was not acted upon until the competition between Beta and VHS formats was resolved. Once the conflict was resolved, there was a rapid and dramatic increase in total market size. The unwillingness of the public to embrace either format was an emergent property of the indirect effects of the competition. Price and quality of the formats had little influence on market size. Instead market size was limited by consumers' fears that they would lose their investment if they made the wrong choice.

Discontinuities are most dramatically visible among nonlinear emergent properties. A physical example occurs in the stalling of an aircraft. At small angles, the lift and (indirectly) alti-

tude of the aircraft increase with the inclination of the wing. At larger angles, however, turbulence will be generated above and in front of the wing causing it to abruptly lose lift.

A particularly remarkable result from nonlinear science is that

... certain seemingly simple natural nonlinear processes, for which the laws of motion are known and completely deterministic, can exhibit enormously complex behavior, often appearing as if they were evolving under random forces rather than deterministic laws [Campbell 06].

In particular, although these processes are deterministic and produce predictable results, they can remain for long periods in intermediate states with unpredictable detail before the predictable outcomes emerge. This phenomenon is known in nonlinear science as deterministic chaos. For an example, consider unfrozen water at 0°C. As heat is removed, the temperature does not change. Instead, the water begins to freeze in unpredictable patterns. It is nevertheless predictable that if heat continues to be removed then all the water will become ice.

Because emergent properties cannot be localized to a single node or constant number of nodes, they are sometimes called *global properties*. Using this term is especially appropriate when contrasting emergent properties with local properties or characteristics of their constituents.

### 3.5 Coherent Structure

Another factor that influences emergence is the natural tendency toward structure and order. A particularly dramatic example is the Red Spot of Jupiter, which emerges from a highly disordered background to exhibit great regularity in its motion. Another example is the structure of the giant ocean waves known as tsunamis. Structure begets structure—as can be seen on a crowded sidewalk where everyone seems to be blocking everyone else. But, once there is a critical mass of flow, others will join in and very quickly the congestion gives way to efficient sequences of pedestrian flows.

In agent-based computer simulations, nearly random actions often result in highly structured emergent behavior. For a simple example, consider John Conway's *Game of Life*, a cellular automata game in which the life and death of a cell is determined by extremely simple rules that depend only on how many of its neighboring eight cells are populated [Gardner 70]. Regardless of how random the initial configuration, regular patterns emerge in the form of shapes or behaviors.

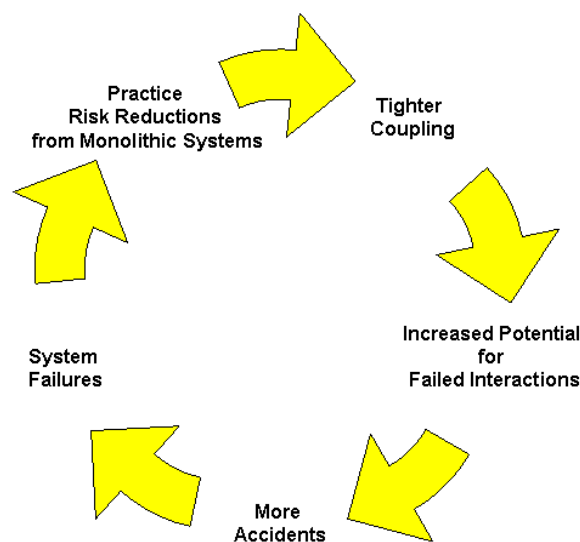
### 3.6 Tight Coupling

In complex systems where emergent effects are prevalent, there is a tendency toward a high risk of accidents. An *accident* is any unintended event that damages subsystems or the system as a whole to the extent that the intended output must be halted promptly [Perrow 99, p. 70].

Accidents arise not as a natural consequence of the complexity but rather from attempts to manage the complexity. Perrow attributes the high risk of accidents in complex systems to unexpected interactions among multiple failures—that is, to unanticipated indirect influences. His larger point, however, is that the more tightly coupled the system, the more likely failures are to influence each other—resulting in a higher risk of accidents. This is a general principle that applies to all systems of systems: the more tightly that components are coupled and their actions and interactions constrained, the more likely that failures will occur and the less likely that intended global properties will emerge.

With or without failures, emergent behavior is ever more constrained as coupling increases. In the presence of local failures, accidents are likely to emerge. Even without failures, tighter coupling reduces the options available for adaptation and evolution. More to the point, any unnecessary coupling will negatively affect feasible solutions in ways such as increased costs, greater resource consumption, or delay. The emergent effects of unnecessary coupling can, in essence, easily preclude all feasible solutions. Thus, even in the absence of accidents, tight coupling can ensure that a system of systems is unable to satisfy its objectives.

Unfortunately, the actions taken to reduce risk in systems of systems typically come from traditional software and systems engineering methods that do not account for emergent effects. These actions—such as adding reporting requirements, imposing more synchronization or test points, increasing coordination, requiring greater visibility, and demanding stronger controls—tighten coupling. Although they may sometimes have beneficial short-term or local effects in a system of systems, approaches that tighten coupling increase risk and further undermine the likelihood of overall success from a long-term, system-wide perspective (Figure 4). In systems of systems, to minimize overall risk, these tradeoffs between local and global goals and between short- and long-term goals must be resolved in favor of the global and long-term ones.



*Figure 4: The Vicious Cycle of Tight Coupling*

## 3.7 Semantic Issues

Wherever people develop, manage, use, or evolve systems, they do so by exerting influence, either directly or indirectly, in the form of communication. Communication involves the transmission of symbols with the attendant semantic issues of the intended meaning and actual interpretations of those symbols. The effectiveness of communication depends on how well interpretations match intended meaning.

One general approach is to codify all relevant semantics and to require that all who wish to participate learn the code. This has been the approach in the semantic Web. The semantic Web idea derives from communication among people with a shared language and culture where there is a broad base of both tacit and implicit shared knowledge built up over long periods of time. Another closely related and successful approach to semantics occurs in specialized knowledge domains where experts in that domain are able to communicate with great efficiency in the jargon of the domain to obtain benefit not obtainable from everyday language. It is unclear that the semantic Web and related approaches can obtain analogous benefits without tacit knowledge and an enormous learning investment by every user.

In many real situations inside and outside automated systems, those who must communicate have different levels of expertise in the domain of interest. In such situations, it is infeasible for the inexperienced person to obtain the level of the expert before communication can commence. For example, when one visits a medical doctor, the symptoms must be communicated to the doctor, not in the language of the medical professional but in the language of the patient. In any communication between inexperienced person and expert, it is the responsibility of the expert to translate in both directions and to continue the interaction until they can come to a common understanding. An automated agent can have only limited effectiveness in providing expert knowledge unless it can interpret inquiries and explain answers in the language of the typical user. Semantics among those with specialized knowledge in different domains is a central problem in systems of systems where users, developers, and managers must be able to communicate, negotiate, and make tradeoffs about issues that arise from expert knowledge in their respective domains.

Semantic issues can also arise dynamically between constituents, especially in joint operations and dynamic network situations where neighbors are not known beforehand. Consider a control device with buttons for forward, back, left, and right. If the buttons are unlabeled or mislabeled, one can quickly determine their functions through experimentation or validation. Semantic issues are important to individual interactions, and in general the shared ontology can be very local in time in space. Most human ontologies do not have to be widely distributed, broadly shared, or even codified.

Traditionally, semantic issues have been handled exclusively in the human domain. There is a dearth of automated semantic methods. Three apparent barriers to their development are the inability of automated systems to (1) learn from their environment, (2) reason correctly beyond the bounds of closed systems, and (3) accurately process incomplete information. The semantic Web provides a data-centric view that separates data from limitations imposed by

specific applications using that data and provides a common application-independent access method. It enables sharing of data independent of application but does not provide automated reasoning or understanding of that data. A Web ontology is a shared taxonomy that classifies terms in a way useful to a specific application domain in which all participants share similar levels of understanding. A Web ontology can assist humans in interpreting automated data, but it does not enable automated reasoning or understanding.

### 3.8 Implications for Emergence from Physics

An important property of biological systems is that each constituent consumes resources at a rate that is less than linear with respect to their number of constituents. In a biological system, local actions and neighbor interactions of each individual autonomous entity may involve use of resources in the form of food, materials, time, number of immediate neighbors, or amount to be remembered. The quantity of each resource used by an individual entity must be bounded by a constant that is independent of the total number of entities in system. In any natural system for which resource consumption exceeds this limit, the cost of participation to individual entities would increase without bound as the size of the system grows, eventually becoming unaffordable and causing the system to fail. Scalability is a necessity for survivability but is achievable only when all costs per entity are strictly less than proportional to the size of the system. (Bounded by a constant is a safe but unnecessarily restrictive approximation.)

Automated systems of systems are similar. An automated system cannot remain scalable or survivable without near-linear bounds on its total resource consumption. That is, no matter how large the system is or may become, its **emergent properties must arise at a cost per constituent that grows less than linearly with the size of the system**. Whether the resource is measured in computational cycles, storage capacity, communications bandwidth, power consumption, dollars, number of defects fixed, or otherwise, this resource limitation must apply. This effect can be seen in certain peer-to-peer (P2P) networks in which each participating member must provide storage proportional to the total membership. In such networks, more may join as the benefits become apparent. But, because the cost to each member grows as each new member is added, at some point the cost of continued membership for an individual becomes unaffordable and members withdraw. Arguments that the P2P network benefits grow linearly with the number of members are unconvincing, in the same way that no one would accept an annual doubling of Internet access fees because the number of Internet users doubles each year. Similarly, the number of cars on a highway at rush hour will not, in the long run, rise above a certain level of saturation, because delay during congestion is at least linear with respect to the number of cars and drivers learn when it is advantageous to find alternative routes. Enough drivers will choose other routes until congestion is nearly balanced on all alternative routes.

That view contrasts with some traditional views that defined scalability as the ability to add new components within some preset limit. As valuable as such definitions may be in some specialized contexts, they permit variability only up to some constant size and thus conflict

with scalability as implying capability for growth to arbitrary size. Note also that such systems have a constant bound on cost, namely the cost of the maximum size supported.

**Emergent behavior cannot be understood from a statistical perspective.** Statistically insignificant local actions can have profound system-wide emergent effects. Consider the integrated circuit. It has had profound impact on the world. No statistical analysis could have predicted either the invention of the integrated circuit or its impact, yet both were predictable by other means. The invention of the integrated circuit was an isolated, statistically insignificant event. At the time of its invention, many who manufactured and used transistors realized that transistors needed to be smaller and that making them smaller required putting all parts of the circuit on a single substrate. That Jack Kilby and Robert Noyce independently but simultaneously invented the integrated circuit is just one indication that its invention was inevitable at that particular time. That the integrated circuit would have enormous importance was immediately understood. The invention of the integrated circuit was also a predictable and inevitable emergent property of the world of electronics in 1958, because the need was widely recognized in the industry at that time and there were no theoretical barriers. The invention of the integrated circuit, however, was also a statistically insignificant event executed by individuals of previously noncritical importance.

Delay has profound implications for the accuracy of information. Knowledge and information are derived through aggregation and transformation of data from multiple sensors and sources. Varying delays in sensors, storage, and communications ensure that the items of data being aggregated are from different points in time. Thus, information can never be accurate unless it is sufficiently imprecise that the time differences do not affect the outcome. For an extreme example that illustrates the effects of communication delay on aggregated information, consider the patterns of stars called constellations. Because there can be thousands of years of delay in the light reaching our eyes from them, the patterns of stars we see do not represent their relative positions today. More to the point, because the light from each star is delayed by an amount proportional to our distance from it, the patterns we see do not correspond to an actual configuration of the stars at any time in history. On a smaller scale, the same must be true of any aggregated information. For example, the common operational picture (COP) envisioned by the U.S. armed services involves aggregation of information from widely dispersed sources with varying degrees of delay, precision, and accuracy. It is possible to create a widely shared view, but it is not possible to guarantee the accuracy of that picture. A difficulty with everyone having the same view is that any inaccuracy in that view will be amplified through its broad, and possibly implicit, support to become a system-wide property. By this means, emergent effects can turn even minor inaccuracies into major failures. In contrast, if each node had a (possibly) unique picture based on its best available information, emergent effects would tend to drive out local inaccuracies that were inconsistent with neighboring information.

**Delay is a critical aspect of emergence.** Every action and interaction contributing to an emergent property consumes time. Thus, emergent effects always occur later than their causes. This may be obvious, but it is often ignored. In the development and use of systems,



there has been a tradition of assuming the absence of communications delay and asserting requirements for zero delay. Zero delay is physically impossible. Quite to the contrary, as systems become more and more geographically distributed, delay increases. Unlike other quality attributes such as bandwidth, reliability, computing speed, and precision of sensors—all of which probably can be improved indefinitely with sufficient investment—delay time is inherently proportional to distance and is limited by the speed of light. The impact of delay becomes an increasingly greater factor in systems as technologies advance, networking increases, and systems become more geographically distributed.

**Emergent properties can be strongly influenced by a shared approach among the constituents.** There is a square in Edinburgh, Scotland, which is the center of an annual festival. During the festival, the square would become congested with people, everyone having great difficulty moving about within the square. A solution, however, was found: arrows pointing in a clockwise direction were placed on each side of a pole in the middle of the square. The arrows influenced enough pedestrians that a traffic flow emerged and the congestion was relieved. This story illustrates an important principle of emergence, namely that structure tends toward structure. Often, as above, it is only necessary to provide a catalyst to trigger the emergence of a pervasive structure.

### 3.9 Summary of Emergence

- Emergent properties are characteristics that arise from the cumulative actions and interactions of the autonomous constituents of a system of systems and cannot be localized to any constant number of constituents. They are unavoidable in systems of systems.
- Emergent properties tend not to arise in closed hierarchically structured systems with global visibility and centralized control. They cannot arise in a truly closed system without autonomous components.
- Emergence can be beneficial, harmful, or neutral in its effect. It is the primary mechanism for both success and failure in systems of systems. Success in systems of systems requires management of emergence through cooperation, use of influence, and focus of shared purpose.
- Emergent properties tend to build on themselves with structure begetting more structure but only when the resulting structure is coherent, when sufficiently many nodes contribute, and often when a catalyst exists to trigger their growth.



---

## 4 Interoperation

This section examines some of the implications of the characteristics of systems of systems and emergent behavior for the acquisition, development, and operation of systems of systems. It distinguishes integration as a centrally controlled process from interoperation as a cooperative, distributed process. Because the assumptions underlying traditional integration methods are not valid above the component level in systems of systems, an alternative approach built on a different set of assumptions is needed. Interoperation is discussed as an alternative to integration in systems of systems. Characteristics distinguishing interoperation from integration include underlying assumptions, scope of concern, perspective of participants, kinds of architecture, and degree of interdependencies with other systems. These distinguishing characteristics suggest several guidelines for success in systems of systems. As presented here, these guidelines are neither complete nor detailed. They instead provide ideas that should be instantiated, validated, and perfected in useful tools and methods for interoperation, through further research and development.

### 4.1 Integration vs. Interoperation

*Integration* is the process of composing or combining subsystems to form a unified system. Historically, both subsystems and the integrated system of which they are a part were viewed as monolithic. In the era of systems of systems, the process of combining autonomous systems to form a system of systems is often called *interoperation*. Integration is a hierarchical process using centralized control and global visibility to bring together major subsystems that were designed to work together in a predefined structure with known fixed roles for each component of the system.

Integration methods exploit a variety of traditional assumptions about systems (including systems of systems) regardless of their size or complexity. They assume that systems have clearly defined boundaries, that all relevant information needed for integration is available or easily obtainable, that requirements are known and fixed, that centralized control is an effective mechanism, that functionality and operational effectiveness are determined by the automated and mechanized aspects of the system, and that sustainment and evolution of the system are outside the scope of concern for development. Traditional integration methods rely on these assumptions.

An integrated system is also made possible by freezing its architecture early in its development, explicitly fixing allocation of functionality among the components, and developing components to satisfy specific needs as determined by requirements and functional allocation.

tions. The architectures of traditional systems are based on data flow, control flow, or other topological relationships that are assumed to remain fixed among the components.

Of course, none of the above assumptions was ever entirely valid; but prior to the era of systems of systems, such assumptions were a reasonable approximation that seldom led to serious problems. They also were of tremendous value in simplifying understanding and reasoning about systems and in facilitating their design and implementation. As systems become larger, more network-interconnected, and more dynamic to meet changing circumstances and as their constituents gain increasing autonomy, the traditional assumptions become increasingly unrealistic.

Interoperation involves cooperative interactions among autonomous systems to create systems of systems that satisfy purposes, goals, or mission objectives that are shared by the participating constituents. Interoperation is a cooperative process using distributed control and trust to bring together, in often unanticipated configurations, systems that were separately developed to achieve a common purpose.

To be effective, the methods used to achieve interoperation must be based on realistic assumptions. Interoperation should be built on assumptions that

- systems of systems are interdependent with other systems beyond their boundaries
- the totality of information needed for precise prediction of outcomes is in general unavailable and unobtainable
- requirements are constantly changing and imprecisely known
- centralized control is ineffective
- hierarchical structures create unnecessary vulnerabilities
- outcomes depend not only on the automated and mechanized aspects of systems but also on the decisions and actions of many stakeholders in their acquisition, development, operation, evolution, and sustainment

It should also be assumed that, in general, systems of systems have ill-defined system-level requirements, a need for continued operation in the presence of component failures and through multiple evolutionary cycles, requirements for scalability, and a need to be adaptable to dynamically changing and unanticipated situations.

Like all architectures, the architecture of a system of systems must provide a vision of those aspects of the system that are shared among the stakeholders and constituents. The architecture of a system of systems cannot be based on characteristics that vary dynamically. For example, the architecture of a mobile wireless system cannot be based on the interconnection structure among the nodes; instead, it must be based on invariant properties such as the protocols of interactions among the nodes. Neither can it be assumed that the architecture of a system of systems can be fixed. Like all aspects of a system of systems, the architecture must evolve in response to changing needs and technology.

Ideally, interoperation is a distributed, cooperative process that composes a system of systems into a networked structure from autonomous constituents. A networked structure is a survivable mechanism that enables self-repair and adaptation in rapidly changing situations. Systems with networked structure suffer from the cost and complexity of redundant capabilities and dynamic adaptation, uncertainties assumed not to exist in traditional systems, and emergent effects not present in monolithic systems. Distributed control, networked structure, and cooperative interoperation are necessary in systems with dynamically changing requirements, variable topology of interconnection, automated error recovery, or scalability requirements.

Because they are composed from monolithic components with fixed functionality and predetermined data and control flows, the system-wide properties of traditional systems are highly predictable in the absence of component failures. Outcomes in traditional systems rely on the functionality of individual nodes, their compositional structure, and sometimes the reinforcing and balancing effects of feedback loops. To the extent that traditional systems are devoid of autonomous components, emergent effects are precluded.

Because they are composed of independently acting constituents, emergent effects in the form of influence, indirect effects, emergent composition, cascade effects, and epidemics are instrumental in determining the system-wide properties of a system of systems. Outcomes in systems of systems rely on the local actions and neighbor interactions of individual nodes and the system-wide properties that emerge from the cumulative effects of those actions and interactions. To the extent that individual nodes act independently in their management, operations, and evolution, emergent effects will predominate.

Given the many characteristics that distinguish monolithic systems from systems of systems, interoperation must be a process that is fundamentally different from integration. In combination, those differences—such as the operational independence of components, the requirements for adaptability and scalability, the critical role of emergent effects, and the independent evolution of the components—and the contradiction of assumptions underlying traditional integration processes require engineering approaches and methods that recognize the changing character of systems. The boundary between integration and interoperation, though, is becoming somewhat blurred in practice, as attempts are made to extend and adapt traditional integration techniques to address systems of systems.<sup>1</sup> It should also be noted that traditional assumptions and integration methods remain appropriate for the monolithic components of systems of systems.

## 4.2 Scope of Interoperation

When systems of systems are considered, our view of what constitutes the scope of a system must change. Traditional integration techniques are inadequate for systems of systems because they focus on the constructive aspects of the automated and mechanized portions of

---

<sup>1</sup> The remarks in this technical report with regard to integration refer to what integration historically has been rather than to what it is becoming.

systems. In the context of systems of systems, it is more apparent that the capabilities and limitations of the system and the services it provides are determined not just by the constructive components of the operational system but also by a variety of other influences—including the acquisition process, operational users, cultural context in which the system is developed, environment in which it is operated, involvement of adversaries and third parties in the operational system, and the intervention of unanticipated events. Catastrophic failures are sometimes observed in systems, even though traditional integration methods are rigorously applied. Often these failures are (correctly) attributed to “stovepiping,” failures in the acquisition process, inadequate user involvement, “fighting the previous war,” funding cuts, and other causes beyond the scope of development and integration—all providing evidence that such systems are, in fact, systems of systems for which traditional assumptions are inappropriate and traditional integration methods inadequate.

Interoperation in systems of systems encompasses human participants in acquisition, development, and operation and all autonomous entities in the system’s programmatic and constructive aspects—not just the automated portions of the operational system. This extension in scope is necessary to include all entities that have significant influence on the actual behavior of the system of systems. Many of the issues of concern in acquisition, development, and evolution are analogous to those in the operational system. Because those who acquire, own, develop, manage, use, or evolve systems are stakeholders who influence and are influenced by the system of systems, their intents, expectations, actions, and interactions for and within the system must be understood and addressed. The independence of management, funding, and decision-making (among program managers, developers, operational users, and other stakeholders) ensures that emergent effects will occur within the programmatic, constructive, and operational aspects of a system of systems and thus that the desired interoperability is unlikely if these effects are ignored. The scope of interoperation in systems of systems must include all effects throughout the life cycle that significantly influence outcomes.

### **4.3 Node-Centric Perspective**

When systems are centrally controlled, hierarchically structured, and composed from monolithic components whose detail is known, it is reasonable to view the system from an external system-centric perspective in which the entire structure of the system can be seen. In a system of systems, however, the overall structure is often unknown and constantly changing. Detail for any given constituent is seldom visible throughout the system. The overall system structure is often ad hoc and continuously evolving. Emergent effects that determine the functionality, quality attributes, and, ultimately, the effectiveness of the system as a whole arise from cumulative effects of the local actions and interactions among the nodes, but with minimal dependence on the precise system structure or details of how other nodes operate internally.

Interoperation in systems of systems demands a node-centric perspective in which each constituent views the system from its own individual perspective. For each node, a node-centric perspective captures the current structure of interconnection and interaction with its immedi-

ate neighbors and as much information about its neighbors as is useful and obtainable. Because local actions and neighbor interactions determine the global system-wide outcome but can occur only at the level of individual autonomous nodes of the system of systems, interoperation requires that individual nodes focus on the contributions of their own actions and neighbor interactions to the fulfillment of shared objectives. An individual constituent node need not be concerned with the internal details or interconnection structure of remote nodes on which it neither depends nor can control.

Recognition, even if only tacit, of the inherent need for a node-centric perspective has led to the concept of service-oriented architectures (SOAs), in which the functionality and protocols of the interactions of individual nodes are accessible but other details, such as structure of interconnection and how the functionality is achieved, are intentionally omitted from the architecture. SOAs are appropriate for interoperation in systems of systems because they provide a node-centric perspective, omit detail that is unneeded or unobtainable, and support independent management and evolution in individual constituent nodes. SOAs, however, are inappropriate—at least in many of their current instantiations—for interoperation in systems of systems because they assume an absence of emergent behavior or at least fail to recognize and provide support for managing emergent behavior.

Interoperation is concerned with the effects, including emergent effects, of the actions and interactions of the individuals, organizations, and mechanized components that compose the system or otherwise influence its outcomes and with how well those outcomes fulfill the intended purpose, goals, and mission objectives of the system. Requirements, architectures, designs, and other operational aspects of a system of systems can be most easily viewed and understood from a node-centric perspective because

- Actions and interactions occur only within and between individual nodes.
- Many of the structural characteristics of the system as a whole change too fast to provide an architectural foundation.
- System-wide purposes and contextual considerations (regardless of their strength) nevertheless serve only to influence local actions and interactions.

The architecture of a system of systems must focus on node-specific capabilities or services in a form that can be combined through cooperative neighbor interactions.

Survivable architectures are ideal for systems of systems because they enable adaptability and scalability and provide a node-centric perspective. As with any architecture, a survivable architecture captures the invariant aspects of a system while allowing noncritical aspects to remain unspecified and variable. Unlike traditional architectures for systems with fixed interconnection structures for data flow or control, however, a survivable architecture allows the control structure, data paths, and topology of interconnections to vary dynamically and specifies those properties of a system of systems that are invariant. A survivable architecture specifies the local actions within nodes and the protocols of interactions between nodes (and nothing more). The local actions define the capabilities or services provided by individual nodes and are usually different for each type of node. The protocols of interaction define the

neighbor relationships between nodes and often differ between types of nodes. Because they relate only to a node's capabilities and do not directly depend on the number of nodes or their interconnection structure, survivable architectures are inherently scalable and highly adaptable to changing circumstances. To their benefit, SOAs often combine a survivable architecture with a data-centric view that provides an application-independent data representation. Unfortunately, SOAs also often add requirements for centralized discovery mechanisms and share ontologies that are unachievable or inappropriate for systems of systems.

Architectures and designs for traditional distributed systems attempt to impose hierarchical control and data flow structures on geographically distributed systems. To the extent that issues of fault tolerance, adaptability, and scalability can be ignored, those attempts are sometimes successful. A hierarchical, distributed system architecture with autonomous nodes would be a centrally controlled, hierarchical system of systems, but with the realization that the root and other nonleaf nodes can only influence their descendants in the hierarchy and that such influence would not necessarily be controlling. That is, every autonomous constituent (even in a hierarchical arrangement) can be viewed as a node in a networked structure, with each node potentially influencing and being influenced by its immediate neighbors (including ancestors and descendants in a hierarchy). The presence of autonomy and lack of downward control, however, means that the resulting architectural, design, management, and operational issues are those of systems of systems, not of hierarchical monolithic systems. And again a node-centric perspective is needed.

Design of an individual node of a system of systems combines local actions that provide possibly (portions of) a capability or service needed within the system with a protocol of neighbor interactions that in combination with anticipated actions of sufficiently many other nodes will ensure that certain desired system-wide properties or services emerge. Processes or algorithms that satisfy this combination, whether executed by people, software, hardware, or some combination, are called emergent algorithms.

More precisely, an emergent algorithm is any computation that achieves formally or stochastically predictable global effects, by communicating directly with only a bounded number of immediate neighbors and without the use of central control or global visibility [Fisher 99]. That is, an emergent algorithm produces system-wide properties or services through cooperative local actions and neighbor interactions distributed throughout a system. Note also that emergent algorithms have a survivable architecture.

## **4.4 Contextual Influences and Constraints**

Although all actions and interactions occur at the node level and systems of systems often appear ad hoc and lack fixed structure when viewed from a system-centric perspective, there must be some basis for cooperation and combined achievement among the nodes. Local actions and interactions are influenced by a variety of contextual conditions, not just by their neighbor interactions. Systems of systems are always interdependent with other systems. It is impossible to develop a system of systems, design an emergent algorithm, or make a local



operational decision within an individual node in ways that will enable interoperation to achieve the desired outcomes of a system of systems without considering the implications of shared objectives and external interdependencies. Similarly, every system of systems and every node within a system of systems resides in a variety of contexts that influence the local actions of the constituents and, ultimately, help determine system-wide outcomes and the degree to which purpose, goals, and mission objectives are satisfied.

Contextual influences include larger systems of systems in which the operational system resides. They include the organizational, legal, funding, and cultural contexts in which the system is acquired, developed, deployed, operated, sustained, and evolved. And, each of these may vary. For instance, “local” reward systems specify benefits and punishments for certain behaviors or measures of success. These systems are called local because they relate to individual nodes in isolation. Typically, reward systems derive from the policies, regulations, or goals of “contextual parents” of one form or another. Contextual influences are sometimes called “constraints,” but at times they fail to constrain and may have little influence. To the extent that they contradict one another, moreover, all constraints cannot be satisfied. They also may conflict with a system’s purpose, goals, or mission objectives. This is particularly obvious when there are major “disconnects” between the intended purposes of the context (for example, acquisition and integration of monolithic systems) and of the system (for example, a cooperatively interoperating system of systems).

The larger the number of contextual influences, the more likely they are to

- contradict one another
- be interpreted differently by each participant
- be ineffectual in determining system-wide characteristics

Consistency among contextual influences increases their effectiveness. But in a system of systems, the greater the number of (nonconflicting) influences, the more likely they are to preclude feasible solutions. When contextual influences conflict, neither one is likely to be satisfied for the system as a whole. Thus, as a general rule, it is important to minimize conflicting constraints; hence, the fewer the number of contextual constraints the better. Major exceptions occur with regard to system-wide goals and simplifying contextual constraints.

To be fulfilled, a system’s purpose, goals, and mission objectives must be widely shared and acted upon by those who acquire, develop, operate, and evolve its constituent parts. The independence of management, operations, and evolution of individual constituents precludes the centralized control required for integration; instead, it necessitates cooperative interaction among constituents that share purpose, goals, and mission objectives. Although cooperation is inherently a node-centric activity, it can only be effective relative to shared purpose, goals, and objectives if each cooperating node takes a system-centric perspective in assessing the implications and likely consequences of each of its own local actions and interactions. The purpose, goals, and mission objects serve as contextual (to the individual constituent nodes) influences that enable cooperation, understanding, and emergence of global (i.e., system-wide) properties.

A simplifying constraint is used to aid understanding and reduce the number of variables that must be managed among constituents. In the acquisition of integrated systems, requirements include not only those necessary to fulfill the purpose, goals, and mission objectives but also many others that specify various aspects of how a system will be implemented. The latter are, in fact, engineering tradeoff decisions made at the requirements level often without engineering analysis. They might identify specific architectures, techniques, or standards; dictate a particular configuration; or designate individual subsystems. Requirements of this kind, which specify how a system will be implemented rather than what is to be achieved, seem inappropriate for a system, monolithic or otherwise, where cost-effectiveness is expected. They are unnecessary and there may be more cost-effective alternatives to any one of them. Such requirements, however,

- provide a shared set of constraints on which each stovepiped component development can depend
- limit the number of variables that must be managed and negotiated among component developers
- constrain the search space for feasible solutions

In these ways, they reduce complexity, aid understanding, and increase the likelihood of success.

However, because systems of systems thrive on adaptability, unnecessary constraints of this kind can preclude feasible solutions and prevent static and dynamic local adaptations among nodes that would otherwise produce more cost-effective solutions. They can prevent adaptation to unexpected and unanticipated situations. They may also assume levels of reliability, certainty, and trust that are not justified by reality.

The simplicity and shared understanding that such requirements engender offer benefits to interoperation of systems of systems as well as to integration of monolithic systems. The problem in systems of systems is not the presence of unnecessary requirements, but that they have traditionally constrained adaptability. A possible remedy for systems of systems would be to specify that certain aspects and relationships will not be globally constrained and are guaranteed to be open to local adaptations and tradeoffs. Requirements of this form would contribute to shared understanding and reduction of complexity, while increasing adaptability.

External influences also include the laws of physics, which might better be classified as constraints because they are more controlling. Communication is fundamental to cooperation, emergence, processing, storage, and other aspects of interoperation. The laws of physics require that communication have delay and furthermore that delay in communication be proportional to distance. Thus, as systems became more geographically distributed, the delay in communication increases. At the same time, as the size of and therefore the delay in electronic circuits decrease, processing speeds increase not only in absolute terms but relative to other (geographically distributed) components of the system. Thus, delay is a constraint with ever-increasing importance and must be managed as an inherent problem of large systems.

Other laws of physics also constrain systems of systems and cannot, as is too often attempted, be defeated by formal requirements that conflict with those laws—no attempt to achieve infinite bandwidth, precisely synchronized clocks, or zero delay will succeed, regardless of requirements to the contrary.

## 4.5 Unnecessary Coupling

Coupling refers to the degree to which components of a system depend on one another. If a modification in some aspect of a component requires changes in another component, then the components are coupled with respect to that aspect. Interdependency and coupling are critical to cooperation and fulfillment of purpose and, thus, cannot be totally avoided. Nevertheless, coupling always has undesirable aspects in the presence of change or adaptation because it prevents modifications from being confined to a single node. Coupling also serves to retard emergent effects. In integration of monolithic systems where emergent effects are generally unintended and often undesirable, even unnecessary coupling can be beneficial. In interoperation among nodes of a system of systems, however, adaptation and evolution are critical success factors, and mission fulfillment relies on emergent behavior. Thus, in systems of systems, any coupling that is not essential to cooperation and fulfillment of purpose can be detrimental.

The degree of coupling can be measured by how many nodes will have to make adaptations, directly or indirectly, in response to changes in one of them. For this reason, it is usually preferable to treat standards and protocols as pervasive, pair-wise agreements rather than as system-wide constraints. The Internet, for example, has never required IPv4 as a standard. Instead each neighbor pair is free to choose whatever protocol is locally most cost-effective. In addition to what other merits may argue for IPv4, however, a critical mass of IPv4 usage has made it the preferred choice for most regions of the Internet. Nevertheless, neighbor pairs either independently or in a contiguous region may choose to use alternative protocols, as is sometimes done, without requiring changes elsewhere—that is, the Internet is loosely coupled relative to protocol. If, however, the use of IPv4 were a requirement for participation in the Internet, then a move to another protocol, such as IPv6, would require coordinated change everywhere and would likely be infeasible. Alternatively, the degree of coupling could be measured by the total system-wide expense resulting from a change within one node.

Regardless of how the cost is calculated or what resources are measured, the tighter the coupling among autonomous constituents, the greater the expense. It is thus important for individual nodes to evaluate candidate local changes in terms of system-wide implications. Expense includes not only direct costs incurred by individual nodes but also the value of opportunity gained or lost system-wide. For example, the value of a service lost when a change delays its availability should be attributed to that change.

The techniques below follow from the characteristics of systems of systems and emergent behavior. They are applicable in acquisition, development, operations, and sustainment and should be considered when developing tools and methods to support interoperation in sys-

tems of systems. They encourage loose coupling and the avoidance of unnecessary constraints.

- avoiding false assumptions (False assumptions not only act as constraints but also frequently conflict with goals and objectives.)
- avoiding unnecessary requirements
- delaying decisions and tradeoffs until they are relevant or needed
- avoiding hierarchical structures
- establishing data conventions and standards independently of the processes, applications, or services that use them
- treating traditional optimization criteria as constraints (Local optimization criteria have often been viewed as objectives—for example, minimize storage space or bandwidth usage. In most applications, however, each constituent node has fixed amounts of storage and bandwidth available and once those constraints are satisfied, additional reductions in storage or bandwidth usage only create idle resources.)
- not being overly precise (The greater the precision with which any constraint is specified, the stronger the constraint. If a piece of functionality would be useful when delivered any time between April 15 and June 15 but is specified to be provided by May 15, then two possible unnecessary constraints could result: the consumer may lose a month's productive use or the producer may be forced to incur greater cost to deliver a month sooner.)
- providing known information (Failure to make known precision available can be as detrimental as being unnecessarily precise. If, for example, the supplier has different costs as a function of delivery date, that information can influence the consumer's choice of delivery schedule.)
- delaying precision specifications until information is needed (The precision to which any specification is made should reflect what is needed and known. Because both the need for and availability of information increase with time, the precision of specifications should be continuously increasing. Otherwise, specifications will tend to be unnecessarily precise initially and eventually insufficiently detailed. Ideally, at each point in time, all specifications will include all that is known but never less than what is needed. To achieve this state, decisions must be made soon enough to satisfy each additional need for precision.)
- being accurate (Accuracy has to do with conformity to reality and is distinct from but sometimes confused with precision. A claim that there are between 20 and 45 states in the U.S. is imprecise because it covers a range of 25 values and inaccurate because the actual number of states is not within the range. All inaccuracies should be avoided because they are inconsistent with reality and thus serve as unachievable constraints.)
- avoiding synchronous mechanisms (For entities to synchronize with each other, whether in time or space, the one that is ready first must wait until the other is ready. Asynchronous operation allows each party to maintain scheduling and locations convenient to itself—though often at the expense of additional storage. Synchronization thus imposes an

unnecessary constraint on the entity that must wait—though often with the benefit of providing a more easily understood system from a system-centric perspective.)

- storing data near where it is used (Physically centralized data repositories not only introduce vulnerabilities and delay in communication and storage, but increase the interdependencies, constraints, and inefficiencies on all nodes by imposing system-wide standards or conventions on data of local concern.)
- cooperating without coordinating (The principle of cooperation without coordination says that only through cooperation, as revealed in each neighbor interaction, can any desired outcome be predictably obtained. At the same time, coordination is an unnecessary synchronization with its own constraining effects and their attendant costs.)

## 4.6 Boundaries of Systems of Systems

A system of systems is often characterized as *unbounded* because no single constituent, individual, or organization within or outside the system has visibility into all aspects of the system. A system can be unbounded because it is networked or otherwise part of a larger unseen world that it influences or is influenced by. It can also be unbounded because it contains commercial off-the-shelf (COTS) or other “opaque” components into which there is minimal insight or visibility. Because systems of systems, and in fact all real-world systems, influence and are influenced by larger systems of which they are a part, all systems of systems are unbounded.

As a practical matter—for purposes of management, operations, funding, testing, and the like—some boundary must be assumed. From the perspective of each stakeholder and each constituent of the system, each system of systems has a boundary. The boundary placement may vary among constituents’ perspectives and from time to time for a given perspective, but nevertheless some boundary is generally assumed.

The choice of boundary is somewhat arbitrary. The boundary determines the nodes considered to be within the system. It should include those entities with the greatest influence and can exclude those that are at great distance or otherwise have little influence. Because every system of systems is ultimately influenced by autonomous entities beyond any assumed boundary, nodes within the system must take special note of neighbors that are outside the boundary. In particular, edge nodes should provide an accurate depiction of influences beyond the system at that point. In this way, boundary nodes act as surrogates and conduits for influences that lie beyond the boundary at that point.

## 4.7 Managing Emergent Behavior

Of particular importance in managing emergent behavior is the understanding that emergent results cannot persist unless they are themselves coherent structures. That is, in establishing purpose, setting requirements, or determining mission objectives, the resulting emergent properties or desired end-states should be in a form that is self-stabilizing. As with the exam-

ple of Section 3.4, both liquid water and ice represent coherent structures and thus are easy to preserve once obtained. The unstable and unpredictable state of water at exactly the freezing point would be a very difficult property to maintain. Because most possible goal states are incoherent and difficult to maintain, systems with emergent behavior will tend toward whichever stable state is widely supported by influences throughout the system. It is not necessary, or in general even possible, that the exact path to achieving that state be known ahead of time. Cooperation and influence are effective mechanisms for achieving shared purposes in loosely coupled systems with autonomous nodes.

The presence of autonomous constituents in systems of systems means that control can never be complete. In the absence of complete control, desired overall effects must be achieved through leadership and other forms of influence. Furthermore, the emergent behaviors that determine system-wide outcomes arise not only from the influences of constituents within the system that support the system's purpose and objects but also from opposing influences within the system and from beyond the administrative domain of the system. Constituents that share the same goals and purposes must influence their neighbors to act in ways supportive of those intentions, whether the neighbors are allies or adversaries. Orchestration is influence focused on desired overall effects and is essential to interoperation in systems of systems.

Because it is aimed at system-wide outcomes, orchestration requires broad-based influence that can be achieved only through cascade effects. Epidemics are particularly appropriate for orchestration because, unlike other cascade effects, they are able to influence a large population without having to direct information to particular nodes within the system. Note also that orchestration is useful not only in the operational system but in development and evolution as well.

In general, emergent behavior does not require altruism in the sense of being detrimental to individual constituents, but it does require consistency of actions sufficient to ensure that properties necessary to achieve shared purpose and goals will arise. Orchestration is a means of achieving shared action and apparently unselfish individual concern for shared objectives by convincing others that those actions are supportive and consistent with their own objectives.

## **4.8 Maximize Accuracy/Minimize Constraints**

When adaptability and evolution are important in the presence of emergent behavior, as is the case with systems of systems, it is not just tight coupling that should be avoided but unnecessary constraints of any kind. Any constraint that is unneeded constitutes an inaccuracy about what the resulting system could be and still adequately fulfill its purpose. In the presence of emergent behavior, inaccuracies, like other influences, tend to get propagated throughout the system (of systems). Individual nodes are thus influenced by an accumulation of inaccurate and often inconsistent information from throughout the system—which reduces the likeli-

hood that any emergent properties will arise, let alone those desired. Inaccuracy is the enemy of predictable emergence.

Some means of maximizing accuracy include never being overly precise, delaying precision specifications until they are needed, negotiating and agreeing to ranges of values rather than point specifications, and making specifications conditional or contingent on future events. Accuracy also demands that throughout the scope of a system of systems, decisions be determined by the power of their supporting rationale and not by the political, financial, or other powers of their advocates. It also helps if the rationale for decisions is documented in a form that is accessible and understandable, so that the same decisions are not made repeatedly and, where appropriate, decisions are made consistently throughout the system.

## **4.9 Modeling and Simulation**

Modeling and simulation (M&S) has long been useful in acquisition, development, operations, and training for complex systems. In systems of systems, change is often too rapid to permit building and testing prototypes for each potential change. Building systems before their requirements are validated by users is prohibitively expensive. Emergent behavior is too complex and unpredictable in detail to envision without simulation assistance. M&S offers potential for augmenting—and significantly reducing the cost of—increasingly expensive test and evaluation processes for systems of systems. M&S also offers an alternative to training with actual systems of systems. This option becomes more desirable as systems of systems become larger and more expensive to train on and as the time for training declines. M&S is essential to the acquisition and development of systems of systems.

Models provide a theory of what a system does or is intended to do. Simulations provide an operational means to predict the implications of that theory. To the extent that the model is correct and consistent with the real system, it can be used to validate requirements, predict emergent effects, and train personnel. M&S complements test and evaluation (T&E). While T&E can provide, at great expense, a few accurate experimental measures of the behavior of an actual system, M&S can inexpensively provide many inaccurate measures of how the system behaves. T&E provides the experimental results to validate and improve the theory (in the form of a model). As the accuracy of the model improves, M&S becomes an increasingly useful means to predicting the behavior of the real system.

## **4.10 Trust**

Trust is a measure of the confidence one party has in the integrity or ability of another. Issues of trust arise wherever systems involve human or organizational constituents. Trust is important in systems of systems because each instance of mistrust creates unnecessary constraints.

In the short term, those who must deal with an untrustworthy party may request more than is needed (i.e., overstate requirements) or promise more than can be provided (i.e., overstate capabilities) on the assumptions that the untrustworthy party will underperform or be satis-

fied with less, respectively. That is, neighbors of mistrusted nodes tend to become untrustworthy themselves, creating a cascading effect that can undermine trust throughout the system with corresponding increases in constraints and reductions in the likelihood of success. The countering force is that greater trustworthiness promotes greater trust and makes it easier to find viable solutions.

In the long term, any system will tend to marginalize untrustworthy constituents, replacing them with more trusted ones or otherwise finding alternatives that eliminate the need to deal with the untrustworthy. In a competitive environment, trustworthy suppliers and consumers are preferred while the untrustworthy lose their customers or pay more for poorer service. In the absence of competition, lack of trust drives organizations to turn inward and to seek increasingly stovepiped solutions where they have greater control. Interoperation depends on a high degree of trust among the autonomous constituents.

It should also be noted that trust

- should be specific to particular capabilities
- varies over time
- is not binary

That an organization can be trusted (i.e., possesses and exercises the skills) to conduct a large-scale software project development does not mean that it will be able to maintain equipment in an operational context. Trust should be partitioned and specific to each type of service offered or domain of expertise required.

Trustworthiness can change as an individual or organization learns new skills or loses critical capabilities or as its skills become more or less appropriate to a changing context. Trust must be repeatedly reassessed. Nothing is either completely trustworthy or wholly untrustworthy. Trust should be a continuous (as opposed to discrete) measure of trustworthiness.

## 4.11 Summary of Interoperability

Interoperation has to do with composing systems of systems through cooperative local actions and neighbor interactions of individual autonomous constituents to satisfy their shared purpose, goals, and mission objectives. Interoperation contrasts with traditional integration mechanisms that impose hierarchical structure through centralized control by an entity assumed to have global visibility. The need for interoperation arises in systems of systems because systems of systems have autonomous constituents, display emergent behavior, and are unbounded. In particular

- Traditional integration methods rely on simplifying assumptions that were reasonable approximations when systems were not interconnected, could be understood in their entirety, were hierarchically structured with centralized control, and did not have independently acting components. Traditional methods become increasingly inappropriate as systems of systems violate more and more of those assumptions, while methods that rely on



cooperative interaction and management of emergent behavior become increasingly effective.

- In contrast with integration, the scope of concern for interoperation must include not only the mechanized components of the operational system but also the operational personnel and other entities that influence the system's actual outcomes—whether in acquisition, development, operation, or evolution.
- Interoperation requires a node-centric perspective with focus on the system-wide implications of local actions and interactions.
- The effectiveness of interoperation depends on the degree to which individual constituents share purpose, goals, mission objectives, and other contextual characteristics.
- While in traditional systems unnecessary constraints (often given as requirements) can improve understanding and reduce complexity, in systems of systems they can add complexity and eliminate feasible solutions. Traditional integration practices minimize the number of variables to limit complexity by reducing the number of choices that must be managed. Interoperation minimizes the number of constraints to reduce complexity by adding flexibility and increasing the number of feasible solutions.
- Desired outcomes cannot be dictated or imposed in the interoperation of a system of systems but instead must be orchestrated through influence, cascade effects, and other cooperative and emergent mechanisms.
- M&S is essential for validating requirements, predicting outcomes, and reducing the time and cost of testing in systems of systems.
- The necessity for measuring and managing trust is proportional to the importance of people and organizations in determining the effectiveness of systems of systems.



---

## 5 Recommendations for Follow-On Work

This technical report serves only as an introduction to the concepts of systems of systems, emergent behavior, and interoperation and to the interdependencies among them. It is incomplete in both depth and breadth. Although a variety of applicable principles and methods are discussed, they represent a small portion of those that are relevant to the development, use, and evolution of systems of systems. Principles and promising methods should be identified and adapted from a variety of domains to provide a more complete picture of potentially useful techniques and approaches. Quantitative and qualitative measures are needed to assess the effectiveness of competing methods. The guidelines of Section 4 should be turned into useful tools and methods.

An analytic framework is needed to organize concepts important to interoperation in systems of systems. It should identify the key considerations and influences from a node-centric perspective. It should be broad enough to incorporate competing processes for cost-effective development, evolution, and use of systems of systems. It should serve as a structure for testing, validating, and refining particular techniques and approaches.

While the many topics discussed in this report need to be pursued in much greater depth, there are also many more relevant topics that were not mentioned. The topics listed below are some of the most promising that were beyond the scope of this report.

- Property-based types (PBTs) offer potential solutions to many of the semantic problems that underlie misunderstandings in interactions among constituents [Fisher 04a, Fisher 04b].
- Agent-based simulation provides the potential for accurate M&S of emergent behavior in systems of systems [Christie 03, Colella 01].
- SOA attempts to provide global services through loosely coupled interactions without prior agreements between providers and users. SOA is an architectural approach that is consistent with many of the needs of systems of systems. It involves methods and approaches that may be more generally applicable to systems of systems [Lewis 04a].
- Because Q methodology addresses subjective measures and is able to capture critical outliers and unique aspects that are statistically insignificant, it may be useful in emergent contexts and in particular for discovering purpose and generating guiding principles [Brown 04].
- Survivable architectures offer insights and methods that should be applicable in any context dependent on emergent effects. They are particularly appropriate for mobile, ad hoc,

or other systems that have dynamically changing or unpredictable topologies of interconnection among their constituents [Fisher 99].

- Complexity Theory underlies many of the ideas in NCW and provides insight into complexity and emergent behavior in physical systems, including mutuality of interdependencies, the importance of even small perturbations, and the inherent dependency of the environment [Moffat 03]. Complexity Theory is not directly applicable to systems of systems because it relies on principles of physical science that do not exist for most interactions in systems of systems; nevertheless, it may provide useful analogies.
- Decentralized thinking approaches emergence from the standpoint of how we think about the world and the need to avoid a centralized point of view [Resnick 96].
- Biological and social systems are in fact nonautomated systems of systems and may offer a variety of proven methods and approaches that can be adapted to automated systems.
- The Projective Analysis (PAN) tool models performance, composition, and implementation risks in emergent contexts and appears applicable to several aspects of interoperation [Boxer 05].
- Network-centric systems constitute an abstract concept of what systems of systems could be, with emphasis on communications technology as a primary enabler [Alberts 99, Alberts 03].
- Autonomic computing offers a related perspective in which self-regulating systems are platform independent, always on, and self-adapting to user needs. This combination is intended to reduce the detail and number of operations that people must think about to gain value from a system [Waldrop 04].
- Predictable assembly from certifiable components (PACC) is a property-based approach to predicting the behavior of component-based systems prior to assembly. Although PACC fails to consider emergent effects, unbounded systems, and autonomous entities necessary for systems of systems and considers only the constructive aspects of systems, some of its methods may be adaptable to interoperation in systems of systems. In particular, its emphasis on automated methods to enforce design and implementation standards, use of objective measures for trusted components, and incremental introduction of new methods have analogies that may be useful in systems of systems. Its use of properties and a reasoning framework may also be pertinent to interoperability [Merson 05].
- Perturbation is an essential catalyst and a potential regulator of emergent effects [Fisher 99]. The speed of propagation of emergent effects is limited by the frequency of interactions among nodes. Thus, rapid results depend on frequent interactions even when there is uncertainty about the relevance of particular interactions to desired outcomes.
- Feasibility is a concept that is poorly understood in emergent contexts. Methods are needed for determining whether a system of systems is feasible, identifying feasible solutions, and performing dynamic assessment and validation.
- In the presence of emergent effects, it may be possible to expand the role of adaptability. In addition to satisfying a system's need for dynamic adaptation or flexibility, adaptabil-

ity might play a more general role in improving efficiency, engaging unknown areas, responding to unanticipated events, facilitating scalability, and gaining asymmetric advantage.

The concepts of emergence and interoperation are now appearing under many names and in many forms. Likely, many of these investigations will provide needed insights and useful methods. Caution, however, should be exercised in embracing the details of any method or approach that ignores emergent effects or depends on centralized control, global visibility, hierarchical structures, bounded systems, coordinated interactions, statistically based predictions, infinite trust, zero delay, infinite bandwidth, or synchronous activity.

Interoperation in systems of systems is an increasingly important and exciting area with more opportunities than answers currently.



---

## References

*URLs are valid as of the publication date of this document.*

- [Alberts 99]** Alberts, D. S.; Garstka, J. J.; & Stein, F. P. *Network Centric Warfare: Developing and Leveraging Information Superiority*. Washington, DC: DoD Command and Control Research Program, 1999.
- [Alberts 03]** Alberts, D. S. & Hayes, R. E. *Power to the Edge: Command...Control...in the Information Age*. Washington, DC: DoD Command and Control Research Program, 2003.
- [Boxer 05]** Boxer, P. & Eiger, C. "Taking Power to the Edge of the Organization: Role as Praxis." *The International Society for the Psychoanalytic Study of Organizations 2005 Symposium*. Baltimore, MD, June 17–19, 2005. <http://www.ispso.org/Symposia/Baltimore/2005%20papers/2005Boxertext.pdf>
- [Brown 04]** Brown, M. *Illuminating Patterns of Perception: An Overview of Q Methodology* (CMU/SEI-2004-TN-026). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004. <http://www.sei.cmu.edu/publications/documents/04.reports/04tn026.html>
- [Brownsword 04]** Brownsword, L., et al. *Current Perspectives on Interoperability* (CMU/SEI-2004-TR-009). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004. <http://www.sei.cmu.edu/publications/documents/04.reports/04tr009.html>
- [Campbell 06]** Campbell, D. K., ed. "Historical Background of *Chaos*." *Chaos*. <http://chaos.aip.org/chaos/staff.jsp#intro> (2006).
- [Carney 05a]** Carney, D.; Fisher, D.; Morris, E.; & Place, P. *Some Current Approaches to Interoperability* (CMU/SEI-2005-TN-033). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. <http://www.sei.cmu.edu/publications/documents/05.reports/05tn033.html>

- [Carney 05b]** Carney, D.; Fisher, D.; & Place, P. *Topics in Interoperability: Systems-of-Systems Evolution* (CMU/SEI-2005-TN-002). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. <http://www.sei.cmu.edu/publications/documents/05.reports/05tn002.html>
- [Christie 03]** Christie, A., et al. *Easel Language Reference Manual and Author Guide Version 3.0b3, 2003-02-24*. <http://www.sei.cmu.edu/community/easel/lrm/index.html> (2003).
- [Colella 01]** Colella, V. S.; Klopfer, E.; & Resnick, M. *Adventures in Modeling: Exploring Complex Dynamic Systems with StarLogo*. New York, NY: Teachers College Press, 2001.
- [Fisher 99]** Fisher, D. A. & Lipson, H. F. “Emergent Algorithms—A New Method for Enhancing Survivability in Unbounded Systems,” Volume Track 7. *Proceedings of 32nd Hawaii International Conference on Systems Sciences (HICSS-32)*. Maui, HI, January 5–8, 1999. Digital Object Identifier 10.1109/HICSS.1999.772824.
- [Fisher 04a]** Fisher, D. A. “Humanistic Approach to Knowledge Processing,” 1–9. *Znalosti 2004 Third Annual Knowledge Conference*. Brno, CZE, February 22–25, 2004. Ostrava, CZE: VŠB TU, 2004 (ISBN 80-248-0456-5) (In Czech).
- [Fisher 04b]** Fisher, D. A. “Reasoning with Property Based Types,” 865–869. *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004)*. Innsbruck, AUT, February 16–18, 2004. Calgary, Alberta, CAN: ACTA Press, 2004 (ISBN 0-88986-404-7).
- [Forester 61]** Forester, J. W. *Industrial Dynamics*. Cambridge, MA: MIT Press, 1961.
- [Gardner 70]** Gardner, M. “Mathematical Recreations: The Fantastic Combinations of John Conway’s New Solitaire Game ‘Life’.” *Scientific American*, 223 (October 1970), 120–123.
- [Gladwell 00]** Gladwell, M. *The Tipping Point: How Little Things Can Make a Big Difference*. New York, NY: Little, Brown and Company, 2000.
- [Lewis 04a]** Lewis, G. A. & Wrage, L. *Approaches to Constructive Interoperability* (CMU-SEI-TR-020). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004. <http://www.sei.cmu.edu/publications/documents/04.reports/04tr020.html>



- [Lewis 04b]** Lewis, G.; Morris, E.; & Wrage, L. *Promising Technologies for Future Systems* (CMU/SEI-2004-TN-043). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004.  
<http://www.sei.cmu.edu/publications/documents/04.reports/04tn043.html>
- [Maier 98]** Maier, M. W. "Architecting Principles for Systems-of-Systems," *Systems Engineering*, 1, 4 (1998): 267–284.
- [Merson 05]** Merson, P. & Hissam, S. "Predictability by Construction," 134–135. *Proceedings of the 20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'05)*. San Diego, CA, October 16–20, 2005. New York, NY: ACM Press, 2005.
- [Meyers 05]** Meyers, B. C.; Monarch, I. A.; Levine, L.; & Smith, J. D. *Including Interoperability in the Acquisition Process* (CMU/SEI-2005-TR-004). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005.  
<http://www.sei.cmu.edu/publications/documents/05.reports/05tr004.html>
- [Moffat 03]** Moffat, J. *Complexity Theory and Network Centric Warfare*. Washington, DC: DoD Command and Control Research Program, 2003.
- [Morris 04]** Morris, E.; Levine, L.; Meyers, C.; Place, P.; & Plakosh, D. *Systems of Systems Interoperability* (CMU/SEI-2004-TR-004). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004. <http://www.sei.cmu.edu/publications/documents/04.reports/04tr004.html>
- [Perrow 99, p. 70]** Perrow, C. *Normal Accidents: Living with High-Risk Technologies*. Princeton, NJ: Princeton University Press, 1999.
- [Resnick 96]** Resnick, M. "Beyond the Centralized Mindset." *Journal of Learning Sciences*, 5, 1 (1996), 1–22.
- [Senge 94]** Senge, P. *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York, NY: Doubleday, 1994 (ISBN 0-385-26095-4).
- [Waldrop 04]** Waldrop, M. M. *Autonomic Computing: The Technology of Self-Management (Publication 2003-7)*. Washington, DC: Woodrow Wilson International Center for Scholars, 2004.  
<http://www.thefutureofcomputing.org/Autonom2.pdf>



<b>REPORT DOCUMENTATION PAGE</b>			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE March 2006		3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE An Emergent Perspective on Interoperation in Systems of Systems			5. FUNDING NUMBERS FA8721-05-C-0003	
6. AUTHOR(S) David A. Fisher				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213			8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2006-TR-003	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116			10. SPONSORING/MONITORING AGENCY REPORT NUMBER ESC-TR-2006-003	
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS			12B DISTRIBUTION CODE	
13. ABSTRACT (MAXIMUM 200 WORDS)  This technical report characterizes systems of systems from several perspectives; shows the role of emergent behavior in systems of systems; and introduces interoperability as the domain of development, use, sustainment, and evolution for systems of systems. It argues that the increasing importance of systems of systems was inevitable, emergent behavior is inherent in systems of systems, traditional software and systems engineering methods are inadequate for interoperation of systems of systems, and emergent methods offer a potential for cost-effective and predictable solutions. This report aims to facilitate discussion and reasoning about interoperation within systems of systems by showing some of the interdependencies among systems, emergence, and interoperation. It establishes a sizable but incomplete repertoire of topics, characteristics, and principles that are fundamental to the intersection of systems of systems, emergent behavior, and interoperation.				
14. SUBJECT TERMS emergence, emergent behavior, interoperation, interoperability, system of systems, systems of systems, systems			15. NUMBER OF PAGES 66	
16. PRICE CODE				
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	